MARCH 1987 VOL. 5 NO. 3 $3.95

FOR THE IBM SYSTEMS PROFESSIONAL

PC TECH JOURNAL®



DESKPRO 386

Compaq's Powerful Entry in the 80386 Sweepstakes

COMPATIBILITY/PERFORMANCE UPDATE

MODULA-2 ROUNDUP

0 71486 02166 7 03

NEW

Eureka: The Solver™

Anyone and everyone who routinely works with equations needs Eureka: The Solver

It solves the most complex equations in seconds. Whether you're a scientist, engineer, financial analyst, student, teacher, or some other professional, you need Eureka: The Solver!

Any problem that can be expressed as a linear or non-linear equation can be solved with Eureka. Algebra, Trigonometry and Calculus problems are a snap.

Eureka: The Solver also handles maximization and minimization problems, does plot functions, generates reports, and saves you an incredible amount of time.

X+exp(X) = 10 solved instantly instead of eventually!

Imagine you have to "solve for X," where X + exp(X) = 10, and you don't have Eureka: The Solver. What you do have is a problem, because it's going to take a lot of time guessing at "X." Maybe your guesses get closer and closer to the right answer, but it's also getting closer and closer to midnight and you're doing it the hard way.

With Eureka: The Solver, there's no guessing, no dancing in the dark—you get the right answer, right now. (PS: X = 2.0705799, and Eureka solved that one in .4 of a second!)

How to use Eureka: The Solver

It's easy.

1. *Enter your equation into the full-screen editor*
2. *Select the "Solve" command*
3. *Look at the answer*
4. *You're done*

You can then tell Eureka to

- *Evaluate your solution*
- *Plot a graph*
- *Generate a report, then send the output to your printer, disk file or screen*
- *Or all of the above*

Eureka: The Solver includes

- ☑ *A full-screen editor*
- ☑ *Pull-down menus*
- ☑ *Context-sensitive Help*
- ☑ *On-screen calculator*
- ☑ *Automatic 8087 math co-processor chip support*
- ☑ *Powerful financial functions*
- ☑ *Built-in and user-defined math and financial functions*
- ☑ *Ability to generate reports complete with plots and lists*
- ☑ *Polynomial finder*
- ☑ *Inequality solutions*

*Introductory price—good through July 1, 1987

Some of Eureka's key features

You can key in:

- ☑ *A formula or formulas*
- ☑ *A series of equations—and solve for all variables*
- ☑ *Constraints (like X has to be < or = 2)*
- ☑ *A function to plot*
- ☑ *Unit conversions*
- ☑ *Maximization and minimization problems*
- ☑ *Interest Rate/Present Value calculations*
- ☑ *Variables we call "What happens?," like "What happens if I change this variable to 21 and that variable to 27?"*

All this power for only $99.95!

Equation-solving used to be a mainframe problem, but we've solved that problem.

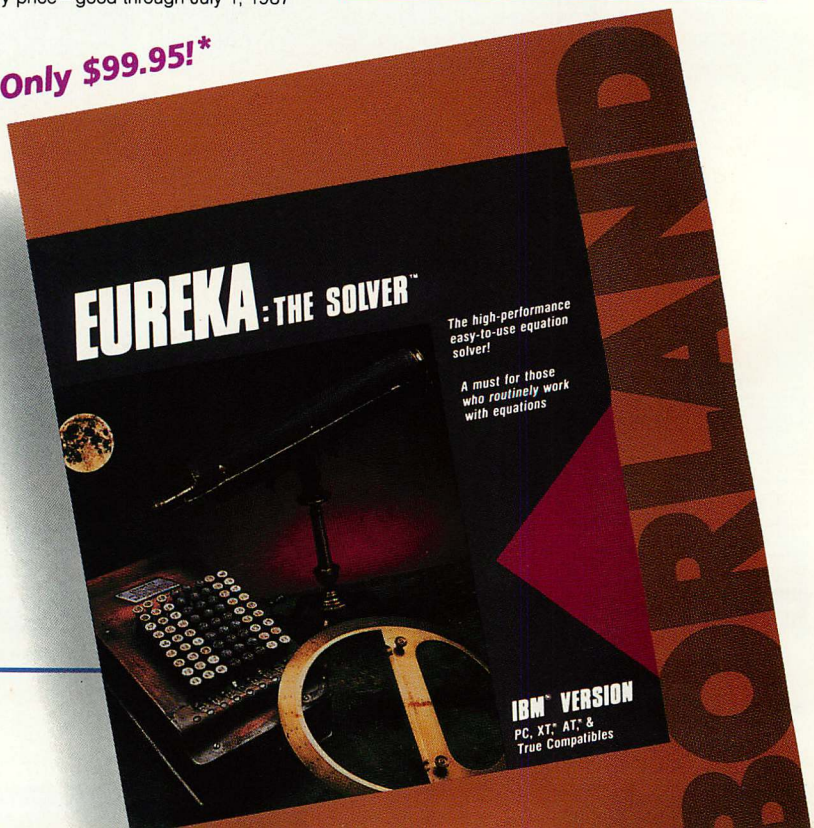
Eureka: The Solver is all you need—and it's yours for only $99.95!

That kind of savings you can calculate with your fingers!

System requirements

IBM PC, AT, XT, Portable, 3270 or true compatibles. PC-DOS (MS-DOS) 2.0 and later. 384K.

Only $99.95!*



BORLAND INTERNATIONAL

Vive la différence

4585 SCOTTS VALLEY DRIVE
SCOTTS VALLEY, CA 95066
(408) 438-8400 TELEX: 172373

GF6

For the dealer nearest you or to order by phone call

(800)255-8008

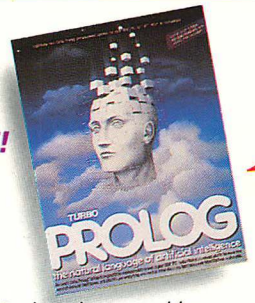
in CA (800) 742-1133 in Canada (800) 237-1136

BI-1103

Turbo Prolog™

> "If you're at all interested in artificial intelligence, databases, expert systems, or new ways of thinking about programming, by all means plunk down your $100 and buy a copy of Turbo Prolog."
> **Bruce Webster, BYTE**

Only $99.95!



Turbo Prolog, the natural language of Artificial Intelligence, is the most popular AI package in the world with more than 100,000 users. It's the 5th-generation computer programming language that brings supercomputer power to your IBM PC and compatibles. You can join the AI revolution with Turbo Prolog for only $99.95. Step-by-step tutorials, demo programs and source code included.

New! Turbo Prolog Toolbox

NEW

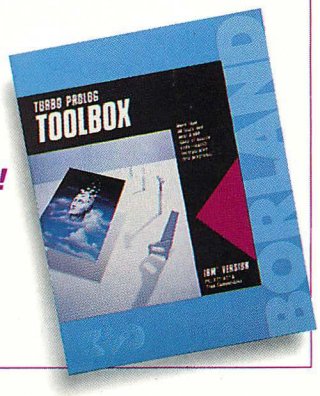
Our new Turbo Prolog Toolbox™ enhances Turbo Prolog—with more than 80 tools and over 8,000 lines of source code that can easily be incorporated into your programs. It includes about 40 example programs that show you how to use and incorporate your new tools.

New Turbo Prolog Toolbox features include:

- ☑ *Business graphic generation*
- ☑ *Complete communications package*
- ☑ *File transfers from Reflex, dBASE III, 1-2-3, Symphony*
- ☑ *A unique parser generator*
- ☑ *Sophisticated user-interface design tools*

It's the complete developer's toolbox and a major addition to Turbo Prolog. You get a wide variety of menus—pull-down, pop-up, line, tree and box—so you can choose the one that suits your application best. You'll quickly and easily learn how to produce graphics; set up communications with remote devices; read information from Reflex,® dBASE III,® Lotus 1-2-3® and Symphony® files; generate parsers and design user interfaces. All of this for only $99.95.

Only $99.95!



System requirements

Turbo Prolog: IBM PC, XT, AT or true compatibles. PC-DOS (MS-DOS) 2.0 or later. 384K. Turbo Prolog Toolbox requires Turbo Prolog 1.10 or higher. Dual-floppy disk drive or hard disk. 512K.

Turbo Pascal®

MACINTOSH™ VERSION ALSO AVAILABLE

The power and high performance of Turbo Pascal is already in the hands of more than half-a-million people. The technically superior Turbo Pascal is the de facto worldwide standard and the clear leader.

The Turbo Pascal family includes:

- *Turbo Pascal® 3.0*
- *Turbo Tutor® 2.0*
- *Turbo Database Toolbox®*
- *Turbo Editor Toolbox®*
- *Turbo Graphix Toolbox®*
- *Turbo GameWorks®*
- *Turbo Pascal Numerical Methods Toolbox™*

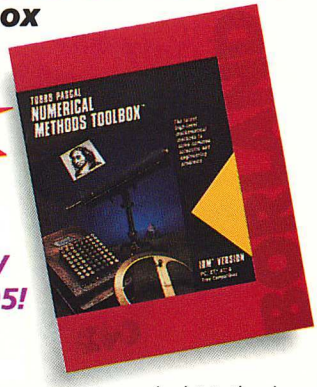


Turbo Pascal, the worldwide standard in high-speed compilers, and family.

> "The language deal of the century."
> **Jeff Duntemann, PC Magazine**

New! Turbo Pascal Numerical Methods Toolbox

NEW



Only $99.95!

What our new Numerical Methods Toolbox will do for you now:

- ☑ *Find solutions to equations*
- ☑ *Interpolations*
- ☑ *Calculus: numerical derivatives and integrals*
- ☑ *Differential equations*
- ☑ *Matrix operations: inversions, determinants and eigenvalues*
- ☑ *Least squares approximations*
- ☑ *Fourier transforms*

As well as a free demo FFT program, you also get Least Squares Fit in 5 different forms:

1. *Power*
2. *Exponential*
3. *Logarithm*
4. *5-term Fourier*
5. *5-term Polynomial*

They're all ready to compile and run.

All this for only $99.95!

System requirements

IBM PC, XT, AT or true compatibles. PC-DOS (MS-DOS) 2.0 or later. Turbo Pascal 2.0 or later. Graphics module requires graphics monitor with IBM CGA, IBM EGA, or Hercules compatible adapter card, and requires Turbo Graphix Toolbox. 8087 or 80287 numeric co-processor not required, but recommended for optimal performance. 256K.

Turbo Pascal 3.0.
Includes 8087 & BCD features for 16-bit MS-DOS and CP/M-86 systems. CP/M-80 version minimum memory: 48K; 8087 and BCD features not available. 128K.

NEW

Turbo Basic®

Introducing Turbo Basic, the high-speed BASIC you'd expect from Borland!

It's the BASIC compiler you've been waiting for. And it's so fast that you'll never have to wait again.

Turbo Basic is a complete development environment; it includes a lightning-fast compiler, an interactive editor, and a trace debugging system.

Because Turbo Basic is compatible with BASICA, chances are that you already know how to use Turbo Basic.

With Turbo Basic your only speed is "Full Speed Ahead"!

You probably already know us for both Turbo Pascal® and Turbo Prolog.™ Well, we've done it again!

We created Turbo Basic, because BASIC doesn't have to be slow.

In fact, building fast compilers is a Borland specialty; both our Turbo Pascal and our Turbo Prolog outperform all their rivals by factors, and with Turbo Basic, we're proud to introduce the first high-speed BASIC compiler for the IBM®PC. If BASIC taught you how to walk, Turbo Basic will teach you how to run!

The Critics' Choice

"Borland has succeeded in stretching the language without weighing us down with unnecessary details . . . Turbo Basic is the answer to my wish for a simple yet blindingly fast recreational utility language . . . The one language you can't forget how to use, Turbo Basic is a computer language for the missus, the masters, the masses, and me.

Steve Gibson, InfoWorld

Borland's Turbo Basic has advantages over the Microsoft product, including support of the high-speed 8087 math chip.

John C. Dvorak"

Turbo Basic ends the basic confusion

There's now one standard: Turbo Basic.

It's fast, BASICA-compatible, and because Turbo Basic is a Borland product, the price is right, the quality is there, and the power is at your fingertips. You see, Turbo Basic's part of the fast-growing Borland family of programming languages—we call it the "Turbo Family." Hundreds of thousands of users are already using Borland's languages, so you can't go wrong. So join a whole new generation of smart IBM PC users—get your copy of Turbo Basic today. You get an easy-to-read 300+ page manual, two disks, and a free MicroCalc spreadsheet—and an instant start in the fast new world of Turbo Basic. All of this for only $99.95—Order your copy of Turbo Basic today!

Free spreadsheet included, complete with source code!

Yes, we've included MicroCalc, our sample spreadsheet, complete with source code, so that you can get started right away with a "real program." You can compile and run it "as is," or modify it.

A technical look at Turbo Basic

- ☑ *Full recursion supported*
- ☑ *Standard IEEE floating-point format*
- ☑ *Floating-point support, with full 8087 (math co-processor) integration. Software emulation if no 8087 present*
- ☑ *Program size limited only by available memory (no 64K limitation)*
- ☑ *EGA and CGA support*
- ☑ *Access to local, static, and global variables*
- ☑ *Full integration of the compiler, editor, and executable program, with separate windows for editing, messages, tracing, and execution*
- ☑ *Compile, run-time, and I/O errors place you in the source code where error occurred*
- ☑ *New long integer (32-bit) data type*
- ☑ *Full 80-bit precision*
- ☑ *Pull-down menus*
- ☑ *Full window management*

System requirements

IBM PC, XT, AT and true compatibles, PC-DOS (MS-DOS) 2.0 or later. One floppy drive, 256K.

Only $99.95!



BI-1103

NEW

Turbo C®

Turbo C: The fastest, most efficient and easy-to-use C compiler at any price

Compilation speed is more than 7000 lines a minute, which makes anything less than Turbo C an exercise in slow motion. Expect what only Borland delivers: Quality, Speed, Power and Price.

Turbo C: The C compiler for amateurs and professionals

If you're just beginning and you've "kinda wanted to learn C," now's your chance to do it the easy way. Like Turbo Pascal, Turbo C's got everything to get you going.

If you're already programming in C, switching to Turbo C will considerably increase your productivity and help make your programs both smaller and faster. Actually, writing in Turbo C is a highly productive and effective method—and we speak from experience. Eureka: The Solver and our new generation of software have been developed using Turbo C.

Turbo C: a complete interactive development environment

Free MicroCalc spreadsheet with source code

Like Turbo Pascal and Turbo Prolog, Turbo C comes with an interactive editor that will show you syntax errors right in your source code. Developing, debugging, and running a Turbo C program is a snap.

Turbo C: The C compiler everybody's been waiting for. Everybody but the competition

Borland's "Quality, Speed, Power and Price" commitment isn't idle corporate chatter. The $99.95 price tag on Turbo C isn't a "typo," it's real. So if you'd like to learn C in a hurry, pick up the phone. If you're already using C, switch to Turbo C and see the difference for yourself.

System requirements

IBM PC, XT, AT and true compatibles. PC-DOS (MS-DOS) 2.0 or later. One floppy drive. 320K.

Technical Specifications

- ☑ *Compiler: One-pass compiler generating linkable object modules and inline assembler. Included is Borland's high performance "Turbo Linker." The object module is compatible with the PC-DOS linker. Supports tiny, small, compact, medium, large, and huge memory model libraries. Can mix models with near and far pointers. Includes floating point emulator (utilizes 8087/80287 if installed).*
- ☑ *Interactive Editor: The system includes a powerful, interactive full-screen text editor. If the compiler detects an error, the editor automatically positions the cursor appropriately in the source code.*
- ☑ *Development Environment: A powerful "Make" is included so that managing Turbo C program development is highly efficient. Also includes pull-down menus and windows.*
- ☑ *Links with relocatable object modules created using Borland's Turbo Prolog into a single program.*
- ☑ *ANSI C compatible.*
- ☑ *Start-up routine source code included.*
- ☑ *Both command line and integrated environment versions included.*

*Introductory price—good through July 1, 1987

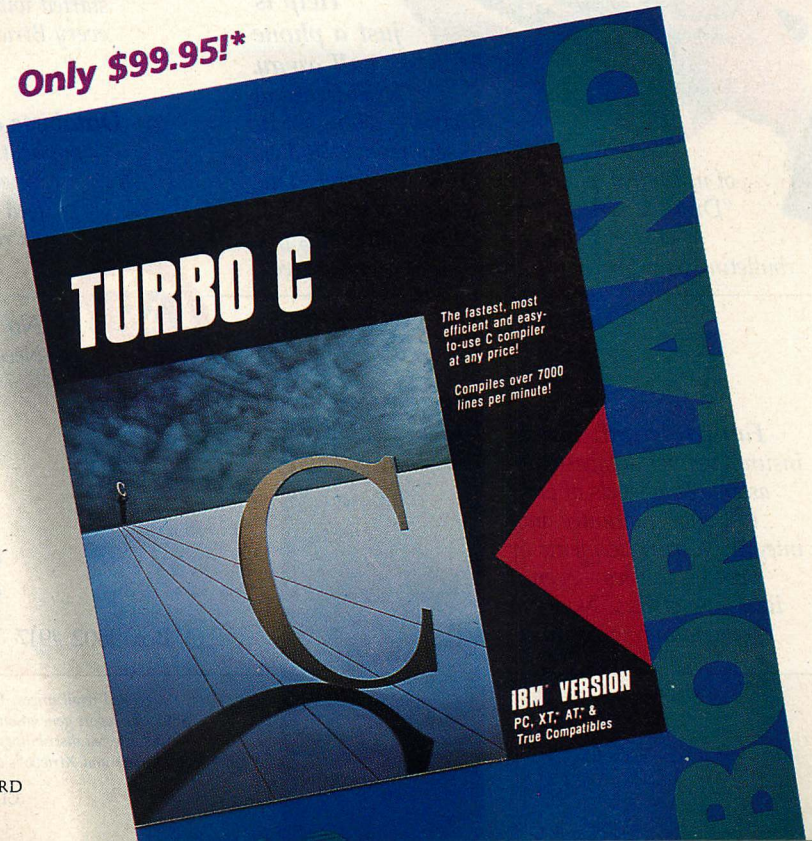
Sieve benchmark (25 iterations)

	Turbo C	Microsoft® C	Lattice C
Compile time	**3.89**	16.37	13.90
Compile and link time	**9.94**	29.06	27.79
Execution time	**5.77**	9.51	13.79
Object code size	**274**	297	301
Price	**$99.95**	$450.00	$500.00

Benchmark run on a 6 Mhz IBM AT using Turbo C version 1.0 and the Turbo Linker version 1.0; Microsoft C version 4.0 and the MS overlay linker version 3.51; Lattice C version 3.1 and the MS object linker version 3.05.

All Borland products are trademarks or registered trademarks of Borland International, Inc. or Borland/Analytica, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders. Copyright 1987 Borland International BI-1103

Only $99.95!*



For the dealer nearest you, or to order by phone

call (800) 255-8008

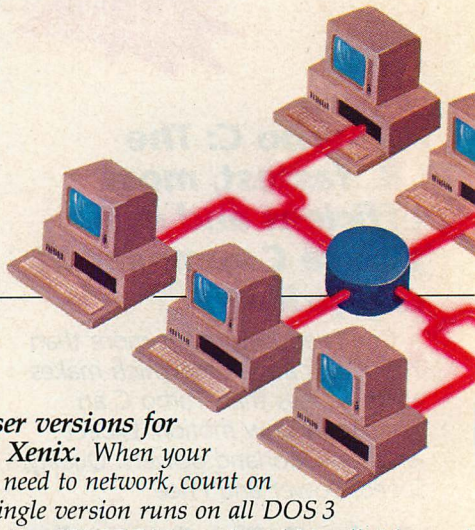
CA (800) 742-1133

Canada (800) 237-1136

CIRCLE NO. 254 ON READER SERVICE CARD

Btrieve®

The Programmer's Choice.



When you're serious about application development, there's just one choice for file management: Btrieve. With what *Computer Language* calls "near mainframe functionality[1]", Btrieve sets the file management standard for PC applications. With Btrieve loaded in your PC, your programs can use simple subroutine calls to retrieve, store and update records.

Multi-user versions for LANs and Xenix. *When your applications need to network, count on Btrieve. A single version runs on all DOS 3 LANs, including IBM PC Network and Novell Advanced Netware. Btrieve is also available for Xenix and multitasking operating systems such as MultiLink Advanced, Microsoft Windows and IBM Topview.*

B-tree based for high performance. *Performance is all-important, especially as your database grows. That's why Btrieve implements the b-tree file structure—the fastest, most efficient method of accessing data.*



Interfaces to C, BASIC, Pascal, COBOL. *Don't waste time programming in awkward fourth generation languages! With Btrieve, simply use the languages you know best—and write applications the right way. Over 15 language interfaces available.*

Built-in security features. *Lock up sensitive data with Btrieve's password protection and unique data encryption scheme—especially useful in local area networks.*





Help is just a phone call away. *Need technical support? You've got it! Btrieve users receive 30 days of unlimited phone support at no charge. This "Direct Connect" policy is renewable for a full year at low cost. And try SoftCraft's free bulletin board for technical tips, seven days a week.*

Thorough documentation, easy implementation. *Getting started with Btrieve is easy: the manual is packed with examples of every Btrieve function in BASIC, Pascal, COBOL and C.*

Database queries, report writing. *Add Xtrieve™ to your Btrieve applications for a fully-relational DBMS. Xtrieve's menu-driven interface gives your users the on-line query capabilities they need—without programming. Add our report writer option to produce custom reports and forms.*





No royalties.
Need we say more?

Fault tolerant. *Btrieve insures against database disasters. Two levels of fault tolerance guarantee data integrity during accidents or power failures—even if lightning strikes. No extra programming required.*



SoftCraft

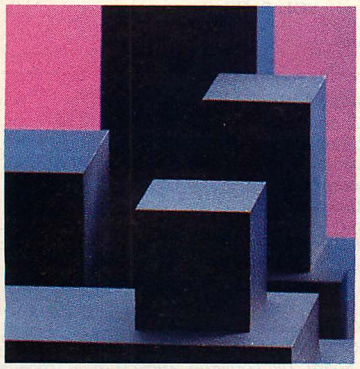
P.O. Box 9802 #917 Austin, Texas 78766 (512) 346-8380 Telex 358 200

Suggested retail prices: Btrieve, $245; multi-user Btrieve, $595; Xtrieve, $245; multi-user Xtrieve, $595 (for report generation, add $145 for single-user and $345 for multi-user). Available from SoftCraft and selected distributors. Requires PC-DOS or MS-DOS 2.X, 3.X, Xenix. Btrieve is a registered trademark and Xtrieve is a trademark of SoftCraft Inc. [1]From Computer Language, *November 1985.*

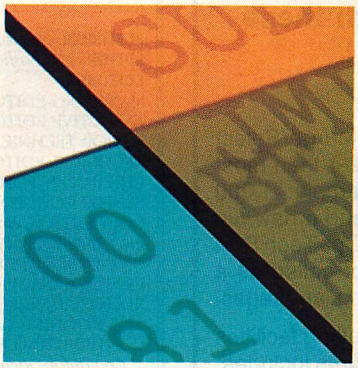
CIRCLE NO. 201 ON READER SERVICE CARD

PC TECH JOURNAL

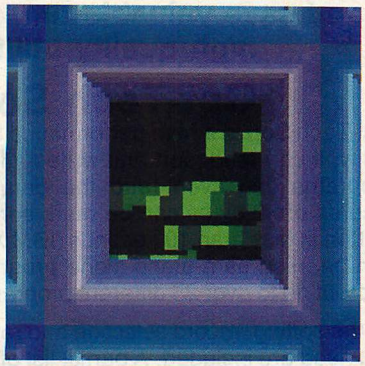
MARCH 1987 VOLUME 5, NUMBER 3



Modular Developments 114



Multilevel Debugger 90



Mapping PC Address Space 102

Compatibility and Performance: **THE NEW STANDARD** / STEVEN ARMBRUST and TED FORGERON
No longer content to wait for IBM, Compaq introduced the Deskpro 386, the first AT compatible to house the Intel 80386. Our tests show that this Compaq is the fastest on the market and holds great promise for the multitasking programs of the future. 48

Compatibility and Performance: **UPDATING THE EVALUATION SUITE** / TED FORGERON, PAUL PIERCE and STEVEN ARMBRUST
To keep up with advancing technology, *PC Tech Journal*'s compatibility and performance metrics have been revised so that they can identify the processor in the test machine—8086/88, 80286, or 80386—and evaluate performance accordingly. 70

MULTILEVEL DEBUGGER / MARK S. ACKERMAN
With CodeView, Microsoft has improved on its line-oriented debuggers, DEBUG and SYMDEB, by offering full debugging capabilities at both the source code and assembly levels. CodeView is packaged with Microsoft's C and FORTRAN compilers. 90

MAPPING PC ADDRESS SPACE / AUGIE HANSEN
Several maps of memory and I/O addresses serve as a guide to inner space for programmers and hardware designers who need to know where the important hardware elements of the PC are located and where new ones can be safely placed. 102

MODULAR DEVELOPMENTS / JOHN T. COCKERHAM
Niklaus Wirth's latest contribution to the field of computer languages boasts the strongest typing and structured syntax yet, along with flexibility and low-level accessibility. Six Modula-2 compilers present a variety of choices for the systems developer. 114

BINARY TRANSFER / RONALD FLORENCE
The XMODEM protocol, considered the norm in PC-to-PC communications, can be adapted to transmit files between DOS and UNIX systems. Code is presented for producing a stand-alone remote XMODEM program for UNIX systems. 144

9 DIRECTIONS
Performance Metrics

15 LETTERS

32 TECH RELEASES

47 TECH NOTEBOOK
XON/XOFF Printer Driver

159 PROGRAMMING PRACTICES
Procedural Algorithms in Prolog

169 PRODUCT WATCH
REPERTOIRE
FILEMAX
Mace Utilities
BASTOC

177 EXPERT CONSULTANT: *APPLIED AI*
What Computers Cannot Do

185 BOOK REVIEWS
Approaching Modula-2

188 TECH MART

196 TECH MARKETPLACE

208 CALENDAR

209 READER SERVICE CARD

Cover photograph · Marc David Cohen

Software Tools

For Programmers & Non-Programmers

Get 'State of the Art' performance and save valuable time with these high quality utilities!

Opt-Tech Sort™

Opt-Tech Sort is a high performance Sort/Merge/Select utility. It can read, sort and write a file faster than most programs can even read the data. Example: 1,000 records of 80 bytes can be read, sorted and a new file written in less than 10 seconds (IBM XT). Opt-Tech Sort can be used as a stand-alone program or called as a subroutine to over 25 different programming languages.

All the sorting, record selection and reformatting facilities you need are included. A partial list of features includes: The ability to process files of any size. Numerous filetypes are supported including Sequential, Random, Delimited, Btrieve, dBASE II & III and many others. Up to 10 key fields can be specified (ascending or descending order). Over 16 different types of data supported. Powerful record selection capability allows you to specify which records are to be included on your output. Record reformatting allows you to change the structure of your output record and to output special fields such as record numbers for use as indexes.

MS-DOS $149. ★ NEW ★ Xenix $249.

★ NEW ★ VERSION

On-Line Help™

★ NEW ★ VERSION

On-Line Help allows you to easily add "Help Windows" to all your programs. On-Line Help is actually two help packages in one. You get BOTH Resident (pop-up) and Callable Help Systems.

The resident version allows you to add help to any system. Your Help System is activated when the "Hot Keys" that you specify are pressed. You can then chain between help windows in any manner you desire.

The callable version allows you to easily display help windows from your programs. A simple call to the help system makes the window appear. The original screen is automatically restored when the help window is cleared. On-Line Help is callable from over 20 different languages.

You have full control over the help window content, size, color and location.

MS-DOS $149. **Demo $10.** (apply toward purchase).

Scroll & Recall™

Scroll & Recall is a resident screen and keyboard enhancement. It allows you to conveniently scroll back through data that has gone off the top of your display screen. Up to 27 screens of data can be recalled or written to a disk file (great for documenting systems operations). Also allows you to easily recall and edit your previously entered DOS commands without retyping.

Scroll & Recall is very easy to use. It's a resident utility that's always there when you need it. MS-DOS $69.

Visa, M/C, AMEX, Check, Money Order, COD or Purchase Orders accepted.

To order or to receive additional information just call and receive immediate highly qualified attention!

Opt-Tech Data Processing

P.O. Box 678 — Zephyr Cove, NV 89448

(702) 588-3737

CIRCLE NO. 222 ON READER SERVICE CARD

PC TECH JOURNAL®

VOL. 5, NO. 3

PUBLISHER: *Newton Barrett*
EDITOR: *Will Fastie*

EDITORIAL
MANAGING EDITOR: *Marjory Spraycar*
EXECUTIVE EDITOR: *Julie Anderson*
SENIOR TECHNICAL EDITOR: *Jim Shields*
TECHNICAL EDITORS: *Caroline Halliday, David Methvin*
CHIEF COPY EDITOR: *Susan Holly*
SENIOR COPY EDITOR: *Gail Shaffer*
COPY EDITOR: *Bruce Ansley*
PROOFREADER: *Elizabeth Wardlaw*
NEW PRODUCTS EDITOR: *Carole L. Eyring*
OFFICE MANAGER: *Trish Ledbetter*
EDITORIAL SECRETARY: *Valerie Rose*
RECEPTIONIST: *Cecilia R. Titus*
CONTRIBUTING EDITORS: *Steven Armbrust, Dave Browning, Michael Covington, Richard M. Foard, Ted Forgeron, Augie Hansen, Thomas V. Hoffmann, Henry F. Ledgard, Ted Mirecki, Max Stul Oppenheimer, Richard Schwartz, Robert Shostak*

ART & PRODUCTION
ART DIRECTOR: *Paula Jaworski*
ASSOCIATE ART DIRECTOR: *Sharon Reuter*
ART ASSISTANT: *Maria Sese*
PRODUCTION MANAGER: *Alison Regan Mrohs*
CONTRIBUTING ARTISTS: *Maciek Albrecht, David Povilaitis*

ADVERTISING SALES
NATIONAL SALES MANAGER: *Rita Burke*
ADVERTISING MANAGER/WEST COAST: *Phyllis Egan*
MARKETING DIRECTOR: *Gayl Sorota*
ASSISTANT TO THE PUBLISHER: *Kathleen Abbott*
ADVERTISING COORDINATOR: *Mary Martin*
MARKETING COORDINATOR: *Kimberly Schroeder*
SALES SECRETARY: *JeanMarie Donlin*
DISTRICT MANAGERS: *Rosemarie Caruso—New England; Arlene Braithwaite—Southeast; Pat Toohey—Mid-Atlantic; Bill Barney—Midwest; Bill Bush, Deborah Gisonni, Nan Hanna—West Coast*
ACCOUNT REPRESENTATIVES: *Polly White—New England/Southeast; Nanette Vilushis—Mid-Atlantic/Midwest; Carey Clarke, Steve Moorman—West Coast; John Blake—National Accounts, Mail Order; Classified advertising director—Kathryn Cumberlander*

CIRCULATION
CIRCULATION MANAGER: *Charles Mast*
CIRCULATION SALES DEVELOPMENT: *Daniel Rosensweig*
MEDIA MANAGER: *Melinda Kendall*
RETAIL SALES MANAGER: *Carol Benedetto*

ZIFF-DAVIS PUBLISHING COMPANY, *a division of Ziff Communications Co.*
PRESIDENT: *Kenneth H. Koppel*
SENIOR VICE PRESIDENT, Marketing: *Paul Chook*
VICE PRESIDENT, Operations: *Baird Davis*
VICE PRESIDENT, Controller: *John Vlachos*
VICE PRESIDENT, Creative Services: *Herbert Stern*
VICE PRESIDENT, Circulation: *Alicia Marie Ivans*
VICE PRESIDENT, Circulation Services: *James Ramaley*
VICE PRESIDENT, Marketing Services: *Ann Pollak Adelman*
VICE PRESIDENT, Development: *Seth Alpert*
VICE PRESIDENT: *Hugh Tietjen*
BUSINESS MANAGER: *Gary A. Gustafson*
PRODUCTION DIRECTOR: *Walter J. Terlecki*

ZIFF COMMUNICATIONS COMPANY
CHAIRMAN: *Philip B. Korsant;* PRESIDENT: *Kenneth H. Koppel;* SENIOR VICE PRESIDENT: *Philip Sine;* VICE PRESIDENTS: *Laurence Usdin, William L. Phillips, J. Malcolm Morris, Steven C. Feinman;* TREASURER: *Selwyn I. Taubman;* SECRETARY: *Bertram A. Abrams*

EDITORIAL OFFICE
PC Tech Journal, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. FAX (group 3): 301/740-8809. MCIMail: PCTECH. PCTECHline: 301/740-8383. Telex: 6502565932 MCI.

ADVERTISING OFFICES
(East Coast/Midwest) Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. 301/740-8300. (New England) 90 Everett Street, Arlington, MA 02174. 617/868-4611. (West Coast) 3460 Wilshire Blvd., Los Angeles, CA 90010. 213/387-2100; 11 Davis Drive, Belmont, CA 94002. 415/598-2290.

SUBSCRIPTION INQUIRIES
PC Tech Journal, P.O. Box 2968, Boulder, CO 80321. Subscription service: 800/525-0643, 303/447-9330. Back issues: send $7/copy ($8 outside U.S.) to Ziff-Davis Publishing, One Park Avenue, 4th floor, New York, NY 10016.

PC Tech Journal (ISSN 0738-0194) is published by Ziff-Davis Publishing Co., a division of Ziff Communications Co., One Park Ave., New York, NY 10016. Published monthly except semi-monthly in December. Subscription rate is $34.97 for one year (13 issues). Additional postage for Canada and Foreign is $8.00/year. Second-class postage paid at New York, NY, and at additional mailing offices. POSTMASTER: Send address changes to PC Tech Journal, P.O. Box 2968, Boulder, CO 80321.

PC TECH JOURNAL is an independent journal, not affiliated in any way with International Business Machines Corporation. IBM is a registered trademark of International Business Machines Corp. Entire contents Copyright © 1987 Ziff-Davis Publishing Company, a division of Ziff Communications Company. All rights reserved; reproduction in whole or in part without permission is prohibited. Direct written requests to Jean Lamensdorf, Licensing Manager, Reprints/Rights & Permissions, One Park Avenue, New York, NY 10016.

BPA

1985 AWARD FOR BEST COMPUTER MAGAZINE Computer Press Association

NEW VERSION 3.0

The Periscope™ Difference

You'll need it sooner or later if you're doing serious software development

When you're writing large and/or complex programs, it's inevitable that you'll have to deal with some tough debugging problems. Your debugger should help you find and solve those problems quickly. The more solid, dependable, and efficient your debugger is, the better able it is to help you out in those difficult situations. The big difference between Periscope™ and other debuggers is this: Periscope enables you to debug programs other debuggers can't handle AND to debug in situations in which other debuggers won't work!

"We have been buying Periscope for about two years now, and have always been more than satisfied with the hardware, the software, and the responsiveness of the company. We have used Periscope in a great many difficult situations, where our only other alternative was a very expensive ICE (in circuit emulator). Periscope has performed most admirably," writes Dr. William Ash, Technical Director, FEL Computing.

The Periscope Promise. Continued product enhancement and user support.

"Not only is your Periscope (Model I) software the greatest thing since K & R, but your support has won over even the heart of this hardened programmer," writes Periscope user Mark Kumler of US Maintenance. *"I had decided long ago that no one in the industry cared about their customers after the check was cashed. You have definitely changed my opinion on that subject!"*

- User ideas are often implemented
- Your first software update is free; later updates are just $20
- You get free technical support and advice
- You can trade up for $10 plus the difference in price
- You get a 30-Day, Money-Back Guarantee

To Order or Receive Free Information, Call Toll-Free:

MasterCard **800/722-7006** VISA

Periscope Quality. The reason for Product of the Month.

"Periscope was chosen as the January (1986) Product of the Month because it represents what we felt was an excellent balance between power and cost and it has an extraordinarily clean and innovative design. . . the overall aura of quality was too strong to ignore," writes Jeff Duntemann, Technical Editor, PC Tech Journal, 7/86.

The Periscope Solution. A full line of debugging products that keep getting better.

"I have used Periscope daily for the past few months for testing and debugging my assembly code and I am still convinced this is the finest hardware or software debugger available at any price," writes long-time Periscope user Wynn Bailey.

There's a model of Periscope to meet your needs and budget. The enhanced Version 3.0 gives you more value than ever before! Call for details.

Periscope I has break-out switch & board with 56K of protected RAM . . . . . $345.

Periscope II has break-out switch . . . . . $175.

Periscope II-X (software-only model) . . . . . $145.

Periscope III has break-out switch & board with hardware breakpoints, a real-time trace buffer, and 64K of protected RAM . . . CALL.

The PERISCOPE Company, Inc.

14 Bonnie Lane, Atlanta, GA 30328 404-256-3860

CIRCLE NO. 215 ON READER SERVICE CARD

Critics Go Hog Wild



Over AutoSwitch™ EGA.



Here's an EGA card to get really excited about.

Autoswitch. "The most versatile EGA board on the market today," that "even the klutziest novice should have up and running in less than 10 minutes." **(PC World 9/86)** That's partly because it has "the best documentation we've seen for such cards." **(Infoworld 7/86)**

But "what really sets Autoswitch apart is its ability to automatically select appropriate video modes from its extensive built-in assortments." **(PC World 9/86)**



In other words, we've "consolidated the best features of other EGA clones into one board, and added an automatic (software) switching feature," **(Infoworld 7/86)** that "work(s) flawlessly." **(Byte 1/87)**

"No board has been more successful at improving IBM's® original idea" **(PC World 9/86)**, and "none simplifies the mechanics and widens the choices as much as Paradise's™ Autoswitch EGA card." **(Personal Computing 9/86)**

And that was before we introduced the **Autoswitch EGA 480 Card** with **132 column mode, 480 vertical line resolution**, and extended our already **unbeatable range of software applications support.**

For the name of your local dealer, call (415) 871-4939.

PARADISE™

AutoSwitch is a trademark of Paradise Systems, Inc.
IBM is a registered trademark of International Business Machines Corporation.

CIRCLE NO. 230 ON READER SERVICE CARD

WHAT'S THE SECRET DEBUGGING WEAPON USED BY EVERYBODY FROM BORLAND TO ORACLE? FREE 44-PAGE ANSWER FROM ATRON.

"WE COULDN'T HAVE DONE IT WITHOUT ATRON'S HARDWARE-ASSISTED SOFTWARE BUGBUSTERS."

Philippe Kahn
Borland Pres.

Larry Ellison
Oracle Pres.

This is the city saved by the Atron bugbusters. Your city. Full of wizards, with hundreds of millions of dollars invested in wringing every ounce of intelligence and performance out of your PC. It used to be plagued with the toughest software bugs known to mankind.

PLAGUES OF BIBLICAL PROPORTIONS

The first and most difficult plague was impossible to trap with software debuggers. These were carnivorous bugs which randomly overwrote programs, data, even the debugger. Nastiest were the ones that slipped in once every few hours, or changed their behavior after each new compile. Forty days and forty nights of recompiling, *trying something else*, caused many a would-be resident of the city to run screaming into the wilderness, never to be heard from again.

Second came the plague of not knowing where the program was, or where it had recently been. This compounded the first plague: How could anyone know *what* caused the random memory overwrites? Add to this random interrupts and timing dependencies, and you begin to understand *The Fear* that gripped the city.

Then came the last plague, which brought the wizards to their knees before they even started debugging. Their towering programs consumed so much memory, there wasn't enough room for their symbol table, let alone debugging software. Even if they could get past the first two plagues, this one killed their firstborn software.

ENTER THE HARDWARE-ASSISTED SOFTWARE BUGBUSTERS

The Atron solution came as a revelation: Monitor every memory reference and every instruction executed, by adding a hardware board to the AT or PC with an umbilical probe to the processor.

The result? Wham! The PC PROBE™ and the AT PROBE™ saved civilization as we know it. The first plague was cured with PROBE'S hardware-assisted breakpoint traps on reading, writing, executing, inputting and outputting. These could be done on single or ranges of addresses, and could include particular data values. All in real time. For a mere software debugger to attempt this, a 1-minute program would take 5 hours to execute.

The second plague, not knowing from whence you came, was cured with PROBE'S real-time trace memory. The history of program execution is saved on-board, in real time. Once a hardware trap has occurred, PROBE displays the program execution in detail, including symbols and source code for C, Pascal, or assembly language programs. Which shows how out-of-range pointers got that way.

The third plague, not enough room for the debugging symbol table to be co-resident in memory with a large program, was cured with 1-megabyte of on-board, hidden, write-protected memory. System memory was then free for the program, keeping the symbol table and debugger safe from destruction.

When the job of bugbusting was done, the wizards used their PROBEs as performance analyzers. So they could have both reliability *and* performance. So they could send only the best software into the field.



IF YOU AREN'T AN ATRON CUSTOMER, ODDS ARE YOU WON'T BE MAKING THE TOP-TEN LIST.

On any given week, at least nine of the top ten best-selling software packages on the Soft-Sel Hotlist come from Atron customers.

Ever heard of Borland? "Without Atron," says its president Philippe Kahn, "there wouldn't be a Side-Kick™, Turbo Lightning™ would be light-years away, and Turbo Prolog™ wouldn't be shipping today."

Ever use a spreadsheet? From Enable™ to Paradox™, their bugs were busted by Atron products.

Into DBMSs? Everyone from Ashton-Tate to Oracle owns at least one Atron bugbuster.

If you use a product from one of the companies in *The City*, you owe life as you know it to Atron. Our guess is that 99% of all PCs, XTs and ATs have at least one product debugged with Atron bugbusters.

FREE 44-PAGE BUGBUSTING BIBLE COULD MAKE YOU A PROPHET, AND YOUR COMPANY A PROFIT.

We've written a complete tutorial on state-of-the-art bugbusting. And it's yours, free for the asking. Full of examples and illustrations, it will show you how the wizards work their magic.

If you're tired of suffering the wrath of program bugs, call Atron today. You could be busting bugs, and sales records, tomorrow.

Atron

THE BUGBUSTERS

20665 Fourth Street • Saratoga, CA 95070 • 408/741-5900

Copyright © 1986 by Atron Corp. PC PROBE™ and AT PROBE™ Atron. The other fine companies mentioned throughout this advertisement own numerous trademarks. Adv. by TRBA.

CIRCLE NO. 203 ON READER SERVICE CARD

Performance Metrics

Measuring computer performance is a tricky business, and benchmarks can be very misleading.

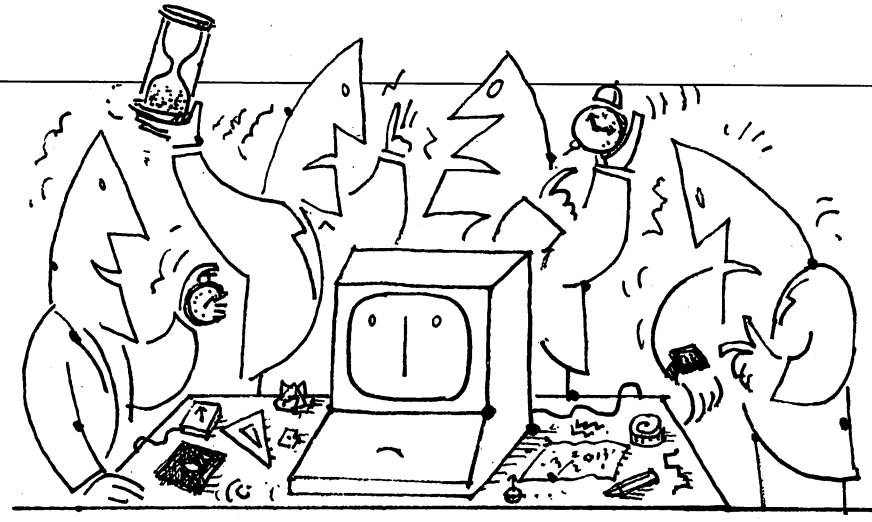


ILLUSTRATION • MACIEK ALBRECHT

Since the day that Peter Norton introduced his now-famous and ubiquitous system information utility, SI, we have all been swept away by a passion for quantifying the performance of the computers on our desks. Our compulsions may be well-founded. Justifying the purchase of advancing technology, is often difficult, especially in the face of plummeting PC prices; the raw performance of the latest model can be a significant selling point.

Norton's SI was a good measure at the time it was introduced, but the hardware technology has moved well beyond what SI is capable of measuring. In fact, it disturbs me that SI continues to be so frequently used, especially by hardware vendors, because it can actually be quite misleading. If the problem were simply measuring 8088 processors that were getting faster and faster, SI would serve perfectly. The fact of the matter is that today's machines might be equipped with any member of Intel's 86-family of processor, from 8088 to 80386; a numeric coprocessor; memory capable of operating with zero wait states, or just plain faster memory; different (from the PC) memory organization; a cache memory system; a higher-bandwidth disk-drive subsystem; a non-AT bus; or an intelligent display adapter. The machine might even have an accelerator product with some combination of the aforementioned features. All these features affect performance in ways that a simple program cannot hope to measure reliably.

SI is not the only benchmark whose contemporary validity I question. Many benchmark programs attempt to mimic the operation of particular types of applications. For example, a benchmark might attempt to measure compute-bound applications or disk-intensive applications or those with some *mix* of the two. Actually, benchmarks that measure processor- or disk-bound performance are usually valid. The problem is that most computer programs have their own, unique fingerprint where the use of system resources is concerned. No "standard" benchmark will ever properly illuminate the performance of unique programs.

To avoid this problem, another type of benchmark measures the performance of a specific program so that its operation may be compared on different computers. The most common is the "typical" Lotus 1-2-3 spreadsheet. This might seem to be a reliable metric because it does indicate how well the machine will fare as a 1-2-3 engine; *PC Tech Journal* uses metrics of this type to measure language compiler performance. However, such tests shed little or no light on the raw performance of the underlying hardware—the very thing we want to measure in our reviews of 80286 and 80386 machines. In fact, this type of benchmark usually masks the performance capabilities of advanced hardware because most programs such as 1-2-3 or compilers are written for the lowest common denominator—the 8088—and, therefore, they may not show an 80386-based machine in its best light even if the observed performance is superior.

Worse, programs written for prior architectures sometimes suffer on the newer machines because they exploit the dark recesses of the older machine's capability; on the newer architecture, some of these "features" become suboptimal. Once again, typical benchmarks usually fail to identify such conditions and any performance degradation becomes invisible if the test computer is faster in other ways.

NEW, IMPROVED!

PC Tech Journal has long recognized this problem. That is why we have taken a different approach in constructing the compatibility and performance metrics that we update in this month's issue ("Updating the Evaluation Suite," Ted Forgeron, Paul Pierce, and Steven Armbrust, p. 70). Our approach is difficult to program, but easy to describe. Rather than taking a single measurement, or reducing a set of observed measurements to a single number, Forgeron, Pierce, and Armbrust have produced an Evaluation Suite that measures the most important contributors to performance and reports the results *individually*. We thus leave the final conclusions to you, because only you can know the specific criteria that bear upon your particular situation and therefore understand which performance issues are germane.

We are going to stop referring to the Evaluation Suite as a set of bench-

marks. The term *benchmark* implies that the answer is somehow absolute and final. I prefer the term *metric*, because we are providing a set of tools that can be used for taking measurements. We consider those measurements excellent in each individual category, but we make no judgment about the performance of the system as a whole because, again, we cannot know the fingerprint.

Our metrics are good, perhaps the best currently available to measure AT-compatible computers. The improvements embodied in the updated code render the suite even more useful than before, although some of the changes are somewhat subtle.

The most significant change is to ATPERF, which now identifies the processor (8088, 8086, 80188, 80186, 80286, or 80386) and selects appropriate measurement techniques for each processor type. This means that the suite can be run on *any* PC and should deliver useful information. Alas, certain measurements cannot be taken on all processor types; in those cases, ATPERF simply does not report a result. One obvious change to the program is the processor-specific code; this was the only way to make valid measurements because of the architectural difference in the processors. ATPERF also has been extended to handle zero-wait-state memory, a feature that is becoming more and more common as hardware vendors attempt to make memory system performance match the rising processor speeds and as the price of faster DRAM chips drops.

READ THE LABEL!

Even though the *PC Tech Journal* compatibility and performance tests do not make judgments or report general indices of performance, each measurement must be understood. An anecdote illustrates the point.

A number of firms use our Evaluation Suite to demonstrate their AT-compatible computers. One demonstrator at Fall COMDEX was showing the difference in performance between his firm's machine and the standard AT. In almost every category, his machine outperformed the AT: However, ATPERF reported that his machine inserted more wait states for graphics board memory accesses than the AT, and he was apologizing to his audience for this, saying that the development team was "looking into the problem."

Of course, there was no problem at all. In each machine the graphics board was the same and *it*, not the processor, was constraining the operation. The graphics board required a fixed amount of realtime to perform its task; in the demonstrator's faster machine, this translated into *more* wait states. Had the demonstrator fully understood the meaning of the metric, he could have avoided the apologies and pointed to yet another indicator of higher performance. Indeed, I explained the meaning, and he changed his pitch.

Understanding each measurement is what allowed us to discover an anomaly in the measurements of the Compaq Deskpro 386. When we ran the program, the times for ROM reads seemed far too high. On closer inspection, it was discovered that the technique used to determine ROM timings (more fully described in the Deskpro 386 review on page 48) were fooled by Compaq's memory organization and the fact that the BIOS is not actually in ROM at the time the measurement is taken. We discovered this problem late in the publication cycle; the article and tables are updated to reflect proper timings, but our published code for ATPERF does not reflect a method for overcoming this particular problem.

This is not an 80386-specific problem. The metric is accurate for every machine we have tested *except* for the Deskpro 386. ATPERF provides accurate measurements on ALR's Access 386 machine, for example. The Compaq experience points out that even our metrics, good as they are, can be fooled by the implementation of techniques designed to improve memory performance. ATPERF's first black eye was the zero-wait-state memory of IBM's XT-286; the Deskpro 386's static column RAM is the second. In the future, cache systems and interleaving will undoubtedly cause similar difficulties. We also are concerned about the hybrid nature of systems with accelerator products; will our tests be valid for an AT with Intel's InBoard 386/AT installed?

The complexity of design and the variety of systems may prevent a general-purpose utility from being written. We hope this is not the case and are working to give ATPERF the ability to handle these new situations. However, special metrics may be needed for special cases, such as the Deskpro 386 and InBoard 386. If so, we will try to provide them.

CALL FOR AUTHORS

As always, we know that among our body of readers are many potential authors for *PC Tech Journal*. I would like to encourage you to get in touch with us if you are interested in writing for our magazine.

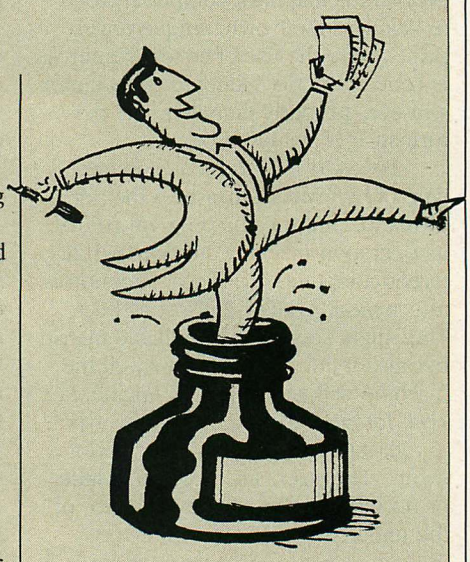
Looking forward in time, we need additional authors who can write on the following topics:

- Artificial intelligence techniques
- Expert systems development
- Data management
- Local area networks
- PC-to-mainframe connections
- PC-to-minicomputer connections
- Programming techniques
- Operating systems

If you would like to be considered for a writing assignment for *PC Tech Journal*, please send a brief résumé outlining your education, experience, and areas of knowledge to:

Marjory Spraycar, Managing Editor
PC Tech Journal, Suite 800
10480 Little Patuxent Parkway
Columbia, MD 21044

We will send you our authors' guide in return. A technical editor will contact you when we need articles related to your areas of expertise.



Your suggestions for articles are also welcome. Our normal process for the development of feature articles involves two stages. We first review a proposal, consisting of an abstract and outline, and then request a first draft. Other material, such as Tech Notebooks and Programming Practices, can be submitted in completed form.

We are looking forward to working with you.

—*WF*

C Programmers!

db_VISTA™: high-speed Database written exclusively for C NOW offers SQL-based Query

"db_VISTA™ has proved to be an all-round high performer in terms of fast execution..."

John Adelus, Hewlett-Packard Ltd./Office Productivity Division

High-speed data retrieval and access... just two benefits of using RAIMA's network model DBMS, db_VISTA. Combine these benefits with those of C—speed, portability, efficiency, and you begin to understand db_VISTA's real measure... performance.

db_QUERY™: new simplicity retains performance!

db_QUERY, our new C-linkable, SQL-based, ad-hoc query and report writing facility ... provides a simple, relational view of db_VISTA's complex network database. No longer will you give up performance for simplicity ... combine db_QUERY with db_VISTA ... you have both!

Independent Benchmark proves High-Speed model 2.76 times faster

An independent developer benchmarked db_VISTA against a leading competitor. Eleven key retrieval tests were executed with sequentially and randomly created key files.

*Result of 11 Key Retrieval Tests

db_VISTA	:671.24 seconds
Leading Competitor	:1,856.43 seconds

db_VISTA's high-speed network database model lets you precisely define relationships to minimize redundant data. Only those functions necessary for operation are incorporated into the run-time program.

Application Portability Complete Source Code

For maximum application portability, every line of db_VISTA's code is written in C and complete source code is available. db_VISTA operates on most popular computers and operating systems. So whether you write applications for micros, minis, or mainframes...db_VISTA is for you.

How db_VISTA works...

Design your database and compile your schema file with the database definition language processor. Develop application programs, making calls to db_VISTA's C functions. Edit and review your database using the Interactive Database Access utility. Compile and link your C program with the db_VISTA run-time library, and your application is ready to run.

Multi-user and LAN capability

Information often needs to be shared. db_VISTA has multi-user capability and supports simultaneous users in either multi-tasking or local area networking environments, allowing the same C applications to run under UNIX, MS-DOS, and VAX VMS.

Royalty-Free Run-Time

Whether you're developing applications for a few customers, or for thousands, the price of db_VISTA cr db_QUERY is the same. If you are currently paying royalties for a competitor's database, consider switching to db_VISTA and say goodbye to royalties.

FREE Technical Support For 60 days

Raima's software includes free telephone support and software updates for 60 days. Technical support personnel are available to answer questions about our software or yours.

30-Day Money-Back Guarantee

Try db_VISTA for 30 days and if not fully satisfied, return it for a full refund.

Price Schedule

	db_VISTA	db_QUERY
☐ Single-user	$ 195	$ 195
☐ Single-user w/Source	$ 495	$ 495
☐ Multi-user	$ 495	$ 495
☐ Multi-user w/Source	$ 990	$ 990
NEW:		
☐ VAX Multi-user	$ 990	$ 990
☐ VAX Multi-user w/Source	$1980	$1980

Call Toll-Free Today!
1 (800) db-RAIMA
(that's 1-800-327-2462)
---OR Call 1-206-828-4636

MasterCard VISA

Read what others say...

"If you are looking for a sophisticated C programmer's database, db_VISTA is it. It lets you easily build complex databases with many interconnected record types. Raima's customer support and documentation is excellent. Source code availability and a royalty-free run-time is a big plus."
Dave Schmitt, President Lattice, Inc.

"My team has developed a sophisticated PC-based electronic mail application for resale to HP customers. db_VISTA has proved to be an all-round high performer in terms of fast execution, flexibility and portability, and has undoubtedly saved us much time and development effort."
John Adelus, Hewlett-Packard Ltd. Office Productivity Division

"On the whole, I have found db_VISTA easy to use, very fast with a key find, and powerful enough for any DBMS use I can imagine on a microcomputer."
Michael Wilson, Computer Language

db_VISTA Version 2.2

Database Record and File Sizes
- Maximum record length limited only by accessible RAM
- Maximum records per file is 16,777,215
- No limit on number of records or set types
- Maximum file size limited only by available disk storage
- Maximum of 255 index and data files

Keys and Sets
- Key length maximum 246 bytes
- No limit on maximum number of key fields per record—any or all fields may be keys with the option of making each key unique or duplicate
- No limit on maximum number of fields per record, sets per database, or sort fields per set
- No limit on maximum number of member record types per set

Operating System & Compiler Support
- **Operating systems:** MS-DOS, PC-DOS, UNIX, XENIX, SCO XENIX, UNOS, ULTRIX, VMS
- **C compilers:** Lattice, Microsoft, IBM, DeSmet, Aztec, Computer Innovations, XENIX and UNIX

Features
- **Multi-user** support allows flexibility to run on local area networks
- **File structure** is based on the B-tree indexing method and the network database model
- **Run-time size**, variable—will run in as little as 64K, recommended RAM size is 256K
- **Transaction processing** assures multi-user database consistency
- **File locking** support provides read and write locks on shared databases
- **SQL-based** db_QUERY is linkable
- **File transfer** utilities included for ASCII, dBASE optional
- **Royalty-free** run-time distribution.
- **Source code** available.

Utilities
- Database definition language processor
- Interactive database access utility
- Database consistency check utility
- Database initialization utility
- Multi-user file locks clear utility
- Key file build utility
- Data field alignment check utility
- Database dictionary print utility
- Key file dump utility
- ASCII file import and export utility

37PT

*The benchmark procedure was adapted from "Benchmarking Database Systems: A Systematic Approach" by Bitton, DeWitt and Turbyfill, December 1983.

RAIMA™ CORPORATION

High-Speed Programming Tools, Designed for Portability

Call Toll-Free Today!

1 (800) db-RAIMA

(that's 1-800-327-2462)

3055-112th Avenue N.E. • Bellevue, WA 98004 USA • (206) 828-4636 Telex: 6503018237 MCI UW

CIRCLE NO. 166 ON READER SERVICE CARD

COMPATIBLE DESIGN. INCOMPARABLE PERFORMANCE. INCREDIBLE VALUES.

CHOOSE FROM THESE COMPLETE SYSTEMS.

PC'S LIMITED TURBO PC™

Monochrome Systems

- Intel 16-Bit 8088-2 System Unit running at 4.77 MHz and 8.0 MHz
- 640K on Motherboard
- AT™-Style Keyboard
- 130 Watt Power Supply
- Hercules Compatible Graphics Adapter with one Parallel Port
- PC's Limited Mono-IIA Flat Screen Monochrome Monitor with Tilt and Swivel Base

with two 360K Floppy Disk Drives—**$859**

with one 360K Floppy Disk Drive and one 20 Meg, 65 MS Hard Disk Drive—**$1199**

EGAds! Color Systems

- Intel 16-Bit 8088-2 System Unit running at 4.77 MHz and 8.0 MHz
- 640K on Motherboard
- AT™-Style Keyboard
- 130 Watt Power Supply
- PC's Limited EGAds! Card
- PC's Limited EGAds! Monitor

with two 360K Floppy Disk Drives—**$1229**

with one 360K Floppy Disk Drive and one 20 Meg, 65 MS Hard Disk Drive—**$1589**

PC's Limited 286[8]

Monochrome Systems

- Intel 80286 running at 8 MHz
- 1024K on Motherboard
- 1.2 Meg Floppy Disk Drive
- Combined Floppy and Hard Disk Controller
- AT™-Style Keyboard
- 192 Watt Power Supply
- Clock/Calendar with Battery Backup
- Hercules Compatible Monochrome Graphics Card
- 2 Serials and 2 Parallel Ports
- PC's Limited Mono-IIA Flat Screen Monochrome Monitor with Tilt and Swivel Base

with 20 Meg, 65 MS Hard Disk Drive, Space Saving Chassis—**$1795**

with 30 Meg, 40 MS Hard Disk Drive, AT™ Standard Chassis—**$2195**

EGAds! Color Systems

- Intel 80286 running at 8 MHz
- 1024K on Motherboard
- 1.2 Meg Floppy Disk Drive
- Combined Floppy and Hard Disk Controller
- AT™-Style Keyboard
- 192 Watt Power Supply
- Clock/Calendar with Battery Backup
- PC's Limited EGAds! Card
- 2 Serials and 1 Parallel Ports
- PC's Limited EGAds! Monitor

with 20 Meg, 65 MS Hard Disk Drive, Space Saving Chassis—**$2195**

with 30 Meg, 40 MS Hard Disk Drive, AT™ Standard Chassis—**$2595**

PC's Limited 286[12]

Monochrome Systems

- Intel 80286 running at 12 MHz
- 1024K on Motherboard
- 1.2 Megabyte Floppy Disk Drive
- Combined Floppy and Hard Disk Controller
- AT™-Style Keyboard
- 192 Watt Power Supply
- Clock/Calendar with Battery Backup
- Hercules Compatible Monochrome Graphics Card
- 2 Serials and 2 Parallel Ports
- PC's Limited Mono-IIA Flat Screen Monochrome Monitor with Tilt and Swivel Base
- AT™-Standard Chassis

with 30 Meg, 28 MS Hard Disk Drive—**$3095**

with 40 Meg, 28 MS Hard Disk Drive—**$3295**

EGAds! Color Systems

- Intel 80286 running at 12 MHz
- 1024K on Motherboard
- 1.2 Megabyte Floppy Disk Drive
- Combined Floppy and Hard Disk Controller
- AT™-Style Keyboard
- 192 Watt Power Supply
- Clock/Calendar with Battery Backup
- PC's Limited EGAds! Card
- 2 Serials and 1 Parallel Ports
- PC's Limited EGAds! Monitor
- AT™-Standard Chassis

with 30 Meg, 28 MS Hard Disk Drive—**$3495**

with 40 Meg, 28 MS Hard Disk Drive—**$3695**



(Tape backup optional)

A GUARANTEE THAT STANDS UP.

PC's Limited has established itself as the leader in PC technology. And the guarantee and warranty programs we offer put us ahead of the pack, too. Here are our terms in plain English.

30-Day Total Satisfaction Guarantee—Any item bought from PC's Limited may be returned within 30 days from the date it was shipped for a full refund of your purchase price. Returned items must be as-new, not modified or damaged, with all warranty cards, manuals, and packaging intact. Returned items must be shipped prepaid and insured, and must bear a PC's Limited Credit Return Authorization (CRA) on the shipping label.

One Year Limited Warranty—PC's Limited warrants the products it manufactures to be free from defects in materials and workmanship for one year following the date of shipment from PC's Limited. During the one year warranty period, PC's Limited will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to PC's Limited.

PC'S LIMITED

To buy PC's Limited computers, call us directly at 1-800-426-5150.

Calls inside Texas, 1-800-252-8336.

1611 Headway Circle, Building 3, Austin, Texas 78754

Sales Calls from anywhere in the country, (512) 339-6962, Technical Support Calls, 1-800-624-9896 or PC's Limited BBS (512) 339-4127

Customer Service Calls, 1-800-624-9897 or MCI MAIL: PC's Limited, Telex No. 9103808386 PC LTD FAX (512) 339-6721

CIRCLE NO. 150 ON READER SERVICE CARD

Now You Know Why BRIEF™ is BEST

"If you need a general-purpose PC programming editor, look no further. Recommended."

- Jerry Pournelle, Byte, 12/86

The Program Editor with the BEST Features

Since its introduction, BRIEF has been sweeping programmers off their feet. Why? Because BRIEF offers the features **MOST ASKED FOR** by professional programmers. In fact, BRIEF has just about every feature you've ever seen or imagined, including the ability to configure windows, keyboard assignments, and commands to **YOUR** preference. One reviewer (David Irwin, DATA BASED ADVISOR) put it most aptly, "(BRIEF)...is quite simply the best code editor I have seen."

WINDOWS

Brief does do windows, and it does them your way!

You can split the screen horizontally and vertically multiple times, creating as many windows as will fit on the screen. Each window can show any part of any file.

BRIEF's flexible, easy to use windows make working with several files a breeze.

"The main thing is that it [BRIEF] will do just about anything you want it to. It has windows— boy, does it ever have windows!...

For the last few months... I've got a raft of mail urging me to try Brief. Now that I've tried it I see why."

Jerry Pournelle - Byte Magazine, Dec. 1986

Every Feature You Can Imagine

Compare these features with your editor (or any other for that matter).

- FAST
- Full UNDO (N Times)
- Edit Multiple Large Files
- Compiler-specific support, like auto indent, syntax check, compile within BRIEF, and template editing
- Exit to DOS inside BRIEF
- Uses all Available Memory
- Tutorial
- Repeat Keystroke Sequences
- 15 Minute Learning Time
- Windows (Tiled and Pop-up)
- Unlimited File Size -(even 2 Meg!)
- Reconfigurable Keyboard
- Context Sensitive Help
- Search for "regular expressions"
- Mnemonic Key Assignments
- Horizontal Scrolling
- Comprehensive Error Recovery
- A Complete Compiled Programmable and Readable Macro Language
- EGA and Large Display Support
- Adjustable line length up to 512

Program Editing YOUR Way

A typical program editor requires you to adjust your style of programming to its particular requirements - NOT SO WITH BRIEF. You can easily customize BRIEF to your way of doing things, making it a natural extension of your mind. For example, you can create ANY command and assign it to ANY key even basic function keys such as cursor-control keys or the return key.

The Experts Agree

Reviewers at BYTE, INFOWORLD, DATA BASED ADVISOR, and DR. DOBB'S JOURNAL all came to the same conclusion - BRIEF IS BEST!

Further, of 20 top industry experts who were given BRIEF to test, 15 were so impressed they scrapped their existing editors!

NOT COPY PROTECTED

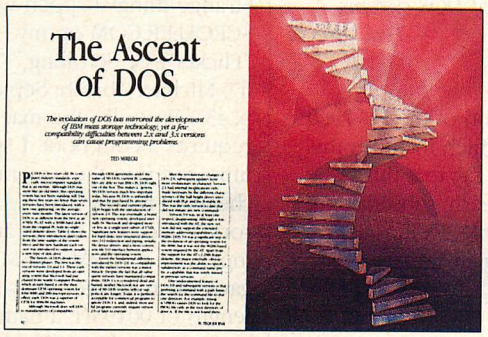
Solution Systems™

MONEY-BACK GUARANTEE
Try BRIEF ($195) for 30 days - If not satisfied get a full refund.
TO ORDER CALL (800-821-2492)

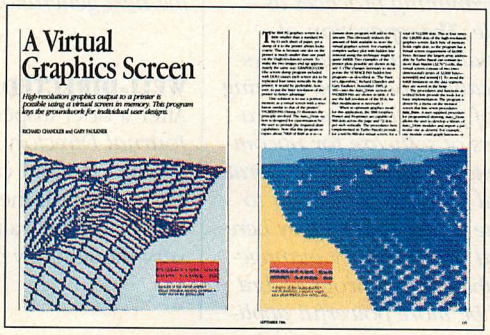
SOLUTION SYSTEMS, 335-P WASHINGTON ST., NORWELL, MA 02061, 617-659-1571

BRIEF is a trademark of UnderWare

CIRCLE NO. 130 ON READER SERVICE CARD


The Ascent of DOS

The evolution of DOS has mirrored the development of IBM mass storage technology, yet a few compatibility difficulties between 2.x and 3.x versions can cause programming problems.


A Virtual Graphics Screen

High-resolution graphics output to a printer is possible using a virtual screen in memory. The program lays the groundwork for individual user designs.

RICHARD CHANDLER and GARY FAULKNER

MYSTERIOUS CHARACTERS

I have been using VSCREEN.PAS from your September 1986 issue ("A Virtual Graphics Screen," Richard Chandler and Gary Faulkner, p. 134), partly to learn something about Pascal.

I am ordering a back issue of the November 1985 *PC Tech Journal* for SURFACE.PAS ("The Painter's Algorithm," Richard Chandler and Gary Faulkner, p. 181) to use with VSCREEN. In the meantime, I have some questions regarding this program. In the past, when I have raised the point of possible typographical errors in printed listings, I have been assured that the listings are printed from actual computer print-outs of tested coding, and therefore, are unlikely to contain typesetting errors. However, my Pascal compiler, Turbo 3.0, hangs up on a line on page 145 of the article:

```
for y :=    to c do
```

It seems that a number is missing from this line, and if I put one in, the compiler proceeds, but I do not know what number should be there.

Also, two end statements appear without semicolons. This is something I had not seen before; however, the compiler complained when I added them, so I kept them out.

I also was unfamiliar with your use of the up arrow character (^) that appeared in several places. Does this have a specific meaning in Pascal?

Finally, in two places (on pages 141 and 142), in a very similar context, a character appears to be smudged in my copy. This character appears after the less-than symbol (<) in the line

```
gotoxy (14,18);
  write ('<    : Begins or Ends line');
```

This is an unusually complex and valuable program to be found in the public domain, and it is greatly appreciated.

Samuel S. Starr
Rose Valley, PA

Thanks to Mr. Starr for pointing out several problems with the program listing for VSCREEN.PAS. I had not noticed them before and one of them is crucial to running the program. The line that Mr. Starr mentions,

```
for y :=    to c do
```

should instead read

```
for y := 0 to c do
```

Turbo Pascal does not allow a semicolon (;) to precede an ***else*** *statement in the* ***if...then...else*** *clause. This is the reason for the missing semicolon following those end statements.*

In Pascal, the ^ (up arrow) character indicates a pointer variable (see chapter 15 in the Turbo Pascal manual). Because Turbo accommodates only a 64KB data segment, the virtual screen had to be stored in the heap, necessitating the use of pointers.

The characters that were not clear in the ***gotoxy (a,b)*** *statements were part of screen directions for these procedures. We were trying to produce something that looked like the symbol on the Enter key. We used the < (60), – (196), and ⊥(217) symbols for this.* PC Tech Journal *has changed the printer used for producing program listings and the software for font selection apparently was not completely bug-free when this listing was printed, thus a strange character (âì) was introduced into the listing. An error similar to this one appears on page 142 in the listing for procedure Write_to_Screen. The remainder of the line that begins*

```
gotoxy (10,12);
```

should actually read as follows

```
write('PgUp/PgDn Scrolls Up or Down ¼
  Screen');
```

We used the ¼ (172) character instead of writing one-fourth and wound up with a tilde (~) in the printed listing.

—Richard Chandler

DOS DISSENT

In his article "The Ascent of DOS" (October 1986, p. 92), Ted Mirecki misses the point about the problems with PC-DOS 3.2, which is, in fact, a radical departure from prior versions. One of the reasons PC-DOS computers are so popular is the ease of installation of most PC-DOS software, which depends on knowledgeable programmers controlling the environment in which their programs run. In version 3.2, IBM/Microsoft took some of that control (the hardware stack size) away from the programmer and gave it to the customer, who ends up calling the programmer to find out why he is getting stack overflow errors, and invariably blames the programmer for the problem. And yet, for all this trouble, version 3.2 offers no operational advantage over 3.1. Software companies should be outraged at IBM for causing such headaches, not to mention the telephone bills, and demand that stack management be returned to the control of the programmer.

James L. Larsen
Computer Consultants
Salt Lake City, UT

As PC-DOS matures and becomes more of a professional business-oriented operating system, it is inevitable that some measure of control is lost. Even in the current state of DOS, with ever more complex interactions between resident programs and a network environment, it is becoming less practical to leave system stack management within application programs. When a program is written, it is impossible to foresee its stack requirements under all of the possible conditions it may encounter. When and if an application fails, it is certainly simpler to tell users to modify a CONFIG.SYS file than to distribute recompiled versions of an application.

It might be argued that taking stack management away from the programmer is not absolutely necessary in

DOS 3.2, but we might as well get used to proper programming practices as soon as possible. The separation of memory, especially stack space, for system and application uses is the defining characteristic of protected-mode operation. Developers of software for upcoming protected-mode multitasking operating systems will have no choice but to relinquish some of the control they heretofore exercised over memory management. Such is the price of progress, but the payoff will be more powerful applications, and even more popularity for the next generation of DOS systems.

—Ted Mirecki

WELL-DIRECTED

After a brief inspection of the *PC Tech Journal Directory*, which includes the product guide, I can say that this issue alone is worth the yearly subscription. It is extremely valuable, well done, and very thoughtfully organized.

Pavel Vladu
Jersey City, NJ

THE ATRON BUGBUSTERS BRING HARDWARE BREAKPOINTS TO MICROSOFT'S CODEVIEW

You already have MicroSoft's CodeView.™ And you've seen our ads for the Atron hardware-assisted software debuggers. Right? You know, the Atron *Bugbusters*? We make the debugging tools used by 9 of the top 10 software developers in the PC market. Now, with our new MiniProbe™ shortcard, you can use your familiar *watchpoints* and *tracepoints* in real time. Without learning new debugging technology.

The MiniProbe can also set a hardware breakpoint over a range of memory locations, helping to trap uninitialized pointers. And MiniProbe has a crash-recovery switch box, which lets you regain control of a frozen system.

Only $395 puts you into world-class debugging.



With real-time watchpoints and tracepoints, a one-minute program will run in one minute. Not 50 hours (the difference between software-only debuggers and hardware-assisted debuggers is a 3000-to-1 increase in efficiency). And if the program bug you're trying to find has anything to do with interrupt activity, it might *never* occur when you're debugging with CodeView alone.

But with the Atron MiniProbe, you can trap events like reading and writing to memory or an IO device. This solves the most common debugging problem: Out-of-range pointers which overwrite the program code or data. Often, the overwrite is different after each new compile of the program.

So now that you don't have to learn a new debugger, the only thing keeping you from debugging like the pros is $395. And our phone number: 408/741-5900. Call today. Bust bugs, and records, tomorrow.

Atron
THE DEBUGGER COMPANY
20665 Fourth Street • Saratoga, CA 95070

© 1986 by Atron. MiniProbe™ Atron. CodeView™ MicroSoft. Atron is a division of Northwest Instruments. TRBA

CIRCLE NO. 249 ON READER SERVICE CARD

BACK TO SCROLLER

Strange things happen when I run SCROLLER.COM on my PC/AT. (See "Flicker-free Scrolling," Tech Notebook 67, Michael Abrash, September 1986, p. 43.) Basically, the machine hangs and requires a cold boot. I spent a lot of time entering the program and double-checking for my own typographical errors. I had zero errors and zero warnings with my IBM MASM 1.0. The linker gave the message "No stack segment," a length of 00193H, a name CSEG, and a program load at 0000:0100. EXE2BIN gave no errors or messages. Could it be a problem with a typo in the article listing, or a conflict with another resident color routine I have, or perhaps it is with the DDIR.COM program (from *PC Magazine*) that I use? Is the program load address correct?

I have an AT bumped to 8 MHz, a Hercules color graphics adapter, and I also load a Norton SA.COM color routine in my AUTOEXEC.BAT file with the ANSI.SYS installed in CONFIG.SYS. The machine boots just fine, the desired colors appear from the SA routine, and DDIR.COM works fine until I load SCROLLER.COM. Then the machine just hangs when I run DDIR.COM.

Flicker is one problem I really have always wanted to resolve. It has always seemed an annoying but simple bug that IBM should have addressed and solved before releasing the original PC and the CGA. This is why I really appreciated your SCROLLER.ASM program when it was published. Thank you for any suggestions you can provide.

William M. Ewers, president
American Progressive Life Insurance
Nashville, TN

Here is one reason (and perhaps a second) why SCROLLER does not work well for Mr. Ewers. The reason his system hangs is that SCROLLER must be converted to a .COM file with the command

EXE2BIN SCROLLER.EXE SCROLLER.COM

then SCROLLER.EXE must be deleted before SCROLLER can be run. Mr. Ewers was running the .EXE file, which is partially overwritten by the next program run (in this case, Mr. Ewers happened to run DDIR next, but any program would have had the same effect). I am afraid that in the interests of saving space I neglected to spell out the correct use of EXE2BIN here.

A possible reason why his screen still flickers after SCROLLER is loaded is that SCROLLER does not eliminate flicker for the CGA, but only for dual-ported

WINDOWS FOR DATA™

The first choice of professional C programmers

"Windows for Data is the best programming tool I've ever used. It's the most flexible I've seen. Whenever I've wanted to do something, I've been able to find a way."

Steven Weiss,
Stratford Systems

Professionals choose our tools because they are designed, crafted, and supported for professionals. Here at Vermont Creative Software, we understand that performance and pleasure in programming derive from more than a long list of functions. **Windows for Data** provides:

PROFESSIONAL FLEXIBILITY: Our customers repeatedly tell us how they've used WFD in ways we never imagined - but which we anticipated by designing WFD for unprecedented adaptability. Virtually every capability and feature can be modified to meet special needs. You will be amazed at what you can do with WFD.

PROFESSIONAL PERFORMANCE: Screen output is crisp and fast. Windows, menus, and data-entry forms snap up and down from the screen. WFD is built upon and includes **Windows for C, the windowing system rated #1 in speed and overall quality** in PC Tech Journal (William Hunt, July 1985).

PROFESSIONAL RELIABILITY: An unreliable tool is worse than no tool at all. VCS products are known in the industry for their exceptional reliability. Ask anyone who owns one.

PROFESSIONAL DOCUMENTATION: Over 600 pages of documentation provide step-by-step explanations for each major application, a reference page for each function, listings of functions alphabetically and by usage, and a fully cross-referenced index. Extensive tutorials and demonstration programs assist learning.

PROFESSIONAL TECHNICAL SUPPORT: The same expert programmers that develop our products provide prompt, knowledgeable technical support.

PROFESSIONAL PORTABILITY: High-performance versions of VCS products are available for XENIX, UNIX, and VMS, as well as DOS. No royalties.

OUR CHALLENGE AND GUARANTEE

If you have an application where no other tool can do the job, try **Windows for Data.** If it doesn't help you solve your problem, RETURN FOR A FULL REFUND. YOU MUST BE SATISFIED.

Ask for **FREE DEMO DISKETTE**



Vermont Creative Software
21 Elm Ave.
Richford, VT 05476
802-848-7738,
Telex: 510-601-4160 VCSOFT

Prices: PCDOS* $295; XENIX, VMS, UNIX Call.
No royalties. Shipping $3.50.
*PCDOS specify C compiler.

WINDOWS FOR DATA

for DOS, UNIX, VMS ...

The complete windowing data entry, menu, and help system that does the hard job others can't — we **guarantee** it!

Pop-up data entry windows; field types for all C data types, plus decimals, dates, and times; auto conversion to and from strings for all field types; system and user supplied validation functions; range checking; required, must-fill, and protected fields; free-form movement; multiple-choice field entry; scrollable sub-forms. Branch and nest windows, forms, and menus.

Complete context-sentitive help system with pop-up windows and scrollable text.

Pop-up, pull-down, scrollable, and Lotus-style menus.

NEW FOR DEBUGGING: Exclusive **VCS Error Traceback System** automatically identifies the location and cause of program errors. Eliminates the need to code error checks on all function calls! **VCS Memory Integrity Checking** helps catch those hard-to-detect, memory-corruption errors.

NEW FOR ERROR HANDLING: Install your own error handler to be called whenever a function detects an error.

NEW FORM LAYOUT UTILITY simplifies form design.

CIRCLE NO. 115 ON READER SERVICE CARD

CGA clones. This means that SCROLLER will not work if the Hercules card is not dual-ported. Mr. Ewers may have been misled by the title for this Tech Notebook, which originally was "Flicker-free Scrolling for CGA Clones." In addition, the introductory blurb should have stated that SCROLLER worked only with CGA clones. I apologize for any inconvenience this has caused.

—Michael Abrash

THE TERMINATOR

In the November 1986 issue of *PC Tech Journal*, the article "Prolog Arrives" (Michael Covington and Andre Vellino, p. 52) gives the following predicate definition that "counts the number of elements in a list. . ."

```
length([],0).
length([H|T],N) :-
    length(T,M),
    N is M + 1
```

and states that "if the terminating condition were written after the recursive part, the program would not terminate." This simply is not the case. This definition can be found in Leon Sterling and Ehud Shapiro's book *The Art of Prolog* (The MIT Press, 1986) on page 131. Either ordering of these rules will work for counting numbers in a list. The terminating condition is a goal such as

```
length([],X)
```

Such a goal will not unify with the head of the recursive rule, hence the recursive rule will not be used. The terminating condition rule would then be tried and the process would halt. In fact, placing the recursive rule first is *more* efficient. The reason is this: consider taking the **length** of a list of size *n*. The recursive rule is going to be used *n* times, whereas the terminating rule is only ever used once. Hence, it makes more sense to put the recursive rule first, saving the interpreter from trying the terminating rule *n* times. Perhaps the authors have in mind that if the recursive definition is placed first, then the definition is not suitable for other than the stated purpose—that is, counting the number of elements in a list, so that, for example, a call such as

```
length(X,2)
```

in which **length** is asked to generate a list of **length** 2, will indeed not terminate as desired.

Richard Denney
Schlumberger Well Services
Austin, TX

Mr. Denney's remarks are correct. Programmers should note, however, that while it is true, in general, that the order of clauses in a Prolog predicate is important, it does not matter in this particular case. This is because when the call to ***length([],M)*** *attempts to bind [] to [H|T], it will fail and try another clause (the program thus will not go into an infinite loop).*

A better example to illustrate the significance of ordering clauses is an alternative definition of ***length*** *in which the terminating condition is slightly different and where* ***cut*** *controls the flow of execution. The program*

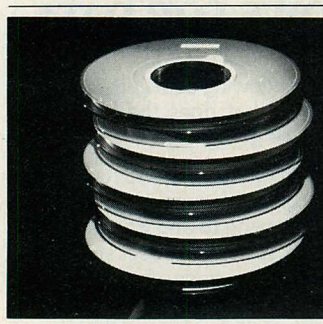
```
length1([X],1).
length1([H|T],N) :-
    !,
    length1(T,M),
    N is M + 1.
```

will terminate successfully; however, the following program construction:

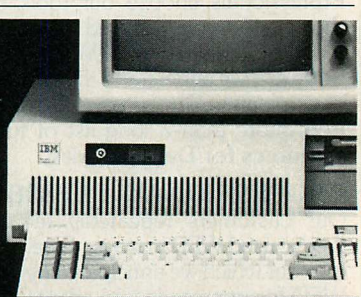
```
length2([H|T],N) :-
    !,
    length2(T,M),
    N is M + 1.
length2([X],1).
```

will fail to terminate.

—Andre Vellino



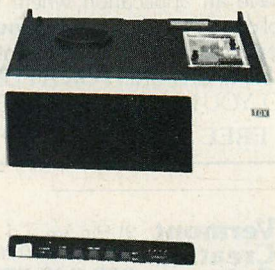
Nothing Should Come Between Mainframe Mag Tapes and Your dBASE or Lotus Except



Telebyte Tape Drives

TDX Mag 9-track 1/2″ Tape Systems from Telebyte provide faster, error-free downloading of mainframe data into your PC. You control the start-stop tape drive either from the keyboard or with Telebyte's exclusive Dataverter software for faster file transfer — the equivalent of a 720,000 bit/second datalink.

Telebyte TDX tape drives are available at either 45 or 75 ips, feature dual density (800/1600 bpi) storage and back up processed files at 2 MB/minute (up to 10 times faster than other 9-track drive systems) as a bonus.



Enter data into dBASE® and Lotus® with no user programming. You do it in two easy steps because Telebyte's exclusive **Dataverter runs under both DOS 2.0 and Xenix™.** Dataverter will automatically convert packed, zoned and unsigned decimal field files, as well as labeled tapes, from EBCDIC to ASCII. The tape system is also supported by software languages in your PC, including C, BASIC, Fortran, Cobol, etc.

Only Telebyte offers such mainframe standards of reliability and IBM-compatible tape drive quality for so little money.

TELEBYTE TECHNOLOGY, INC.

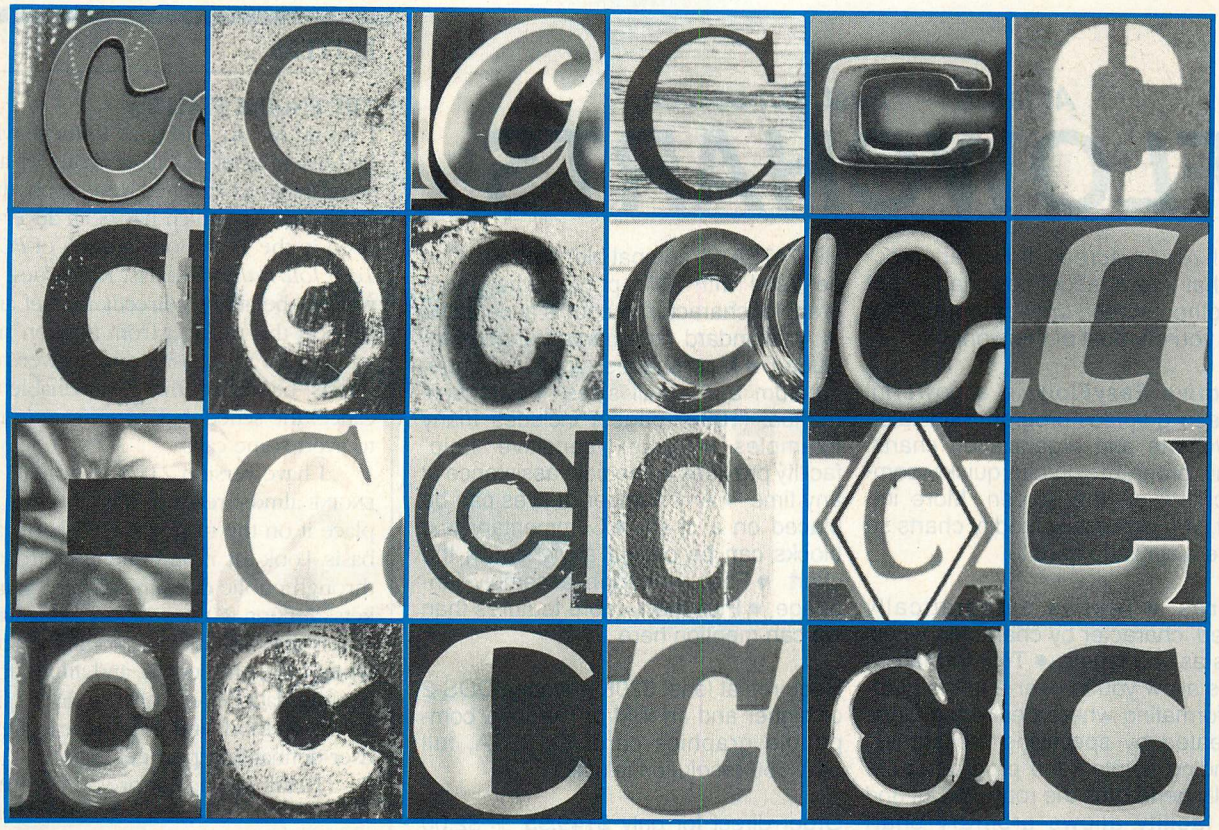
GSA Contract Number GS00K86AGS5301

1-800-835-3298

Telebyte Technology, Inc. • 270 E. Pulaski Road • Greenlawn NY 11740 • (516) 423-3232

dBASE® is a registered trademark of Ashton-Tate, Inc.; Lotus® is a registered trademark of Lotus Development Corporation; IBM® is a registered trademark of International Business Machine Corporation; Xenix™ is a registered trademark of MicroSoft.

CIRCLE NO. 155 ON READER SERVICE CARD



NEW! FROM BLAISE COMPUTING

C Tools Plus™

For The Programmer Whose Alphabet Begins & Ends With "C"

Today's programmers need more than yesterday's tools. Requirements such as removable windows and "sidekickable" pop-up utilities are changing the face of program design. You need to filter interrupts so that other resident programs still work. You need the ability to switch between multiple display pages and monitors. Today's technical demands are almost endless, but C TOOLS PLUS gives you what you need.

SOLID LIBRARY SUPPORT

Blaise Computing offers you solid library support that can meet all your demands and more. C TOOLS PLUS embodies the full spectrum of general-purpose utility functions that are critical to today's applications.

Here's just part of the PLUS in C TOOLS PLUS:

- **C TOOLS and C TOOLS 2** compatibility—two packages that receive rave reviews for quality, organization, usability and documentation.
- **FULL SOURCE CODE**
- **WINDOWS** that are stackable, removable, that support word wrap and that can accept user input.
- **INTERRUPT SERVICE ROUTINE** support for truly flexible, robust and polite resident applications.
- **MULTIPLE** monitor and display support, including EGA 43-line mode.
- **FAST DIRECT VIDEO ACCESS** for efficiency that will not constrain good program design.
- **DOCUMENTATION, TECHNICAL SUPPORT** and attention to detail that have distinguished Blaise Computing products over the years.

C TOOLS PLUS** supports the Microsoft (and IBM) 3.00 and Lattice 3.00 C compilers and is just **$175.00.

Also Available Are:

***C VIEW MANAGER**— A kit for building data entry screens and menus. Begin by designing on-screen what the operator will see; call upon our library functions from your program to display the screens and retrieve the data. Just $275, including all library source code.*

***C ASYNCH MANAGER** — provides the crucial core of hardware interrupt support needed to build applications that communicate. It also includes the "XMODEM" file-transfer protocol and support for Hayes-compatible modems. All source code is included for $175.*

***C TOOLS & C TOOLS 2**— an indispensable combination still available at a low price of $175, including all source code. See review in* PC Tech Journal, *6/85.*

BLAISE COMPUTING INC.

2560 Ninth Street, Suite 316 Berkeley, CA 94710 (415) 540-5441

ORDER TOLL-FREE 800-227-8087!

CA residents call (415) 540-5441

YES, send me the PLUS I need! Enclosed is $_____ for C TOOLS PLUS. (CA residents add 6½%, Sales Tax. All domestic orders add $10.00 for Federal Express shipping.)

Name: _____ Phone: (____) _____

Shipping Address: _____

City: _____ State: _____ Zip: _____

VISA or MC #: _____ Exp. Date: _____

CIRCLE NO. 104 ON READER SERVICE CARD

At last! - Fast, On-screen

FLOWCHARTS

Finally! An on-screen flowchart processor that knows about flowcharts - not just another "screen draw" program that makes you do most of the work.

Interactive EasyFlow is a powerful full-screen graphics program dedicated to flowcharts and organization charts. With this program you can quickly compose charts on the screen. More important, you can easily modify charts so they are always up to date.

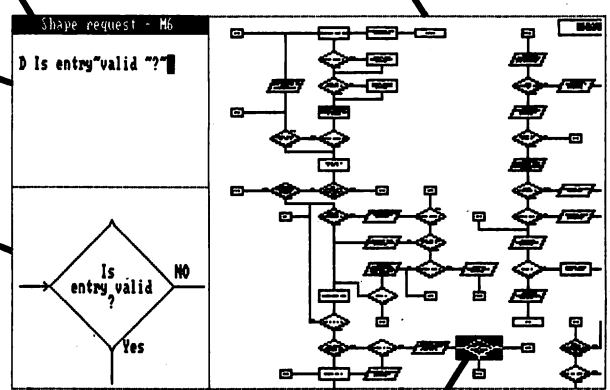
Features: • Text is automatically centered, character by character, within shapes as you type it • Text formatting controls allow you to over-ride the automatic formating where desired • Lines are created by specifying the starting and ending points - the program automatically generates the route • Cut and paste facility allows arbitrary chart fragments to be moved, copied rotated, reflected or sent to/from disk • Shape insert-delete and row/column insert-delete • Charts can be up to 417 characters wide by 225 lines high. Charts too wide for the printer are automatically printed in strips. • Charts can be larger than the screen - the window into the chart scrolls both horizontally and vertically as necessary • Works with many popular matrix printers including Epson, IBM graphics printer and compatibles. Full support for HP LaserJet and LaserJet Plus. Works with HP 7475A (& compatible) plotters. Can be used with ANY printer when non-graphic (character) output is acceptable • All standard flowcharting shapes included • Most shapes supplied in large, medium and small sizes • Extensive manual (100+ pages) includes many examples • Context sensitive "help" facility provides immediate assistance at any time • Any number of titles can be placed on a chart • Commentary text blocks can be placed anywhere in the chart • Fast: written in assembly language • Plus many more features than we can mention here

Requires at least 320K memory, DOS-2 or higher and an IBM or Hercules compatible graphics card. On EGA, full 640x350 resolution is used.

Order direct for only **$149.95** + $2.00 S&H (USA/Canada), $10.00 (foreign). Payment by MO, check, VISA, MasterCard, COD or Company PO. Rush orders accepted ($15.00 S&H; USA/Canada only). Rush orders received by noon will be delivered the next business day (to most locations).

Order Desk: 1-800-267-0668

The sample screen display shown below is typical of what you see while editing a chart. Other screen displays are provided for entering titles, changing options, getting "help" and so on.



HavenTree Software Limited
P.O. Box 1093-N
Thousand Island Park, NY 13692
Information: (613)544-6035 ext 48

CIRCLE NO. 113 ON READER SERVICE CARD

LETTERS

THE ENGINE STALLED

I would like to apologize to those who had the interest to call or write for a software product I had developed, the BASIC ENGINE, which has been advertised in the Tech Book section of *PC Tech Journal* for the last few issues. I will not be able to directly market and support this product from now on and I will not be in a position to answer requests. I am sorry that I was unable to cancel the scheduled advertising in time to avert some calls.

I have version 2.0 of the BASIC ENGINE almost ready to ship, but will place it on the market on a "shareware" basis. Look for it on bulletin boards and through public domain software markets. Version 2.0 has a more enhanced editor that will allow screening for data types, case, and length, and allows formatting data better; it has an index file access method; and it has an excellent user-definable report generator.

Thank you again for your interest.

David A. Violette
Praxis Software Engineering
Green River, WY

A SUCCESSFUL EXPERIMENT

Back in 1983, *PC Tech Journal* published an article by Richard M. Foard on XB, Experimental BASIC ("The Anatomy and Construction of XB," July/August 1983, p. 61). That article was the most valuable tool that my company and I used for more than 18 months. I compiled your XB precompiler back then and made only minor alterations to increase its speed. It clocked in at about 266 lines per minute and we have used it since that time to develop between 50 and 100 XB-compiled programs with upwards of 2,000 lines each.

Now times have changed and most of the new program development we do is in the C language. However, we still have a lot of maintenance to do on what we have developed over the years, so we still use XB. Some people might believe that the new Microsoft QuickBASIC compiler has reduced the usefulness of XB, but that is not the case. The new compiler balloons the code so much, that we are sticking to our original IBM BASIC compiler. But even at 266 lines per minute, when processing a 2,000-plus line program, XB seems slow. So I have recently rewritten XB in the C language. Our new version of XB clocks in at 666 lines per minute.

Again thank you for the fine coverage of Experimental BASIC.

Douglas Hill
Glendale, MO

MICROSOFT LANGUAGES NEWSLETTER VOL. 2, NO. 3

News about the Microsoft Language Family

Tracing through User Libraries with Microsoft® QuickBASIC

Large programs are easier to maintain and debug if they are divided into smaller, more manageable parts. These units are called modules and may contain up to 64K of subprograms. You may separately compile subprograms with Microsoft QuickBASIC and build user libraries of these BASIC subprograms as well as assembly language routines. These user library routines can be used over and over again and are linked into your BASIC programs as needed.

Debugging these user libraries is made very simple with the built-in debugger in Microsoft QuickBASIC. QuickBASIC's built-in Debugging mode allows you to control the rate of program execution and the number of source lines displayed during execution. The Debugging mode is activated when the TRON statement is executed and is turned off with the TROFF statement. You can activate the Debugging mode for part or all of a source file. The Debugging mode has three "submodes": Step, Trace and Animate.

When you debug your BASIC programs that call subprograms contained in user libraries, Microsoft's QuickBASIC Debugging mode lets you trace through the user library routines when you activate the Trace mode. This mode traces program execution one line at a time like the Step debugging mode but you may also trace program execution through subprograms and user-defined functions. If you call any assembly language routines in your main program, the Trace mode will not trace through each line of the assembly routine but will trace the entire routine as a whole. Because the built-in debugger allows you to display both the source code and the output screen, you may observe the program execute while seeing the output.

Support Available for Microsoft Language Products

Once you purchase your Microsoft language product, a number of support services are available to you. If you have any problems with your product, you may call the Microsoft Product Support Hotline at (206) 882-8089 for assistance. A Product Support Representative will try to help you find a solution. In addition, Microsoft provides technical assistance on some electronic bulletin boards, such as CompuServe®.

If you are a software developer who needs in-depth information, you may purchase Microsoft DIAL, an integrated set of on-line services. You can gain access to information on the DIAL bulletin board (which lists answers to frequently asked technical questions and provides information on new products, bug lists, seminars, and many other topics), and if the answer you are looking for is not on the bulletin board, you can electronically submit technical assistance requests (TARs) directly to Microsoft's support organization. Contact Linda McCarty at (206) 882-8080 for an application form and additional information on DIAL.

Filling out and sending in your product registration card is very important. It will allow you to receive notices about new versions and enhancements to your Microsoft product. The update notices let you know about the enhancements and instruct you on how you may receive the updated version. If you are in the market for a language product, you may receive useful information from this newsletter; you may also purchase a subscription to our Microsoft System Journal that contains technical articles on our languages and operating systems/environments. Microsoft Consumer Response can help you by sending you data sheets and other information such as the Language Support Directory (for available third-party libraries that support Microsoft languages).

For more information on the products and features discussed in the Newsletter,
write to: Microsoft Languages Newsletter
16011 NE 36th Way, Box 97017, Redmond, WA 98073-9717.
Or phone:
(800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

Latest DOS Versions:	
Microsoft C Compiler	4.00
Microsoft COBOL	2.10
Microsoft FORTRAN	4.00
Microsoft Macro Assembler	4.00
Microsoft Pascal	3.32
Microsoft QuickBASIC	2.01

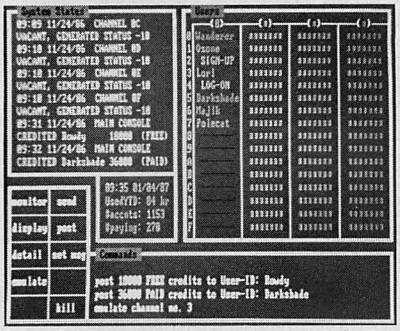
Microsoft is a registered trademark of Microsoft Corporation. CompuServe is a registered trademark of CompuServe, Inc.

Look for the Microsoft Languages Newsletter every month in this publication.

MULTI USER BBS

Off-the-shelf and custom systems for:

★ Multi-User Teleconferencing
★ Multi-User Electronic Mail
★ Multi-User File Upload/Download
★ Multi-User Order Entry
★ Multi-User Games and Amusements
★ Multi-User Database Lookup
★ Multi-User Online Expert Systems
★ Multi-User Catalog Scanning
★ Multi-User Classified Advertising
★ Multi-User Educational Services



What do you need for your Multi-User Bulletin Board System?

	Us	Them
16 modems on one card	YES	?
Up to 64-user capability	YES	?
Runs under MS-DOS V3.1	YES	?
C source code available	YES	?
Menu-oriented operation	YES	?
Accounting w/audit-trail	YES	?
Extensive SYSOP displays	YES	?
Powerfail-protected data	YES	?
"Midnite cleanup" option	YES	?
1-year hardware warranty	YES	?

We sell hardware and software for the IBM PC family and compatibles. Our product line is centered around the GALACTICOMM BREAKTHROUGH, a single-slot card with 16 independent modems on it. You will simply have a cable coming out the back of your machine, going straight into the jacks in the wall installed by the telephone company. No external hardware needed.

Call our multi-user demo system with your modem, at (305) 922-3901. Then call (305) 472-9560, voice, for more information. Why not call right now?

GALACTICOMM

GALACTICOMM, Inc., 11360 Tara Drive, Plantation, FL 33325

CIRCLE NO. 199 ON READER SERVICE CARD

LETTERS

MERGING TRAFFIC

Peter G .Aitken's interesting review of hard-disk cards for the PC in your January 1987 issue ("Mass-Storage Mergers," p. 76) contains an error in the two BASIC programs listed for verification of the ROM BIOS version. For the IBM PC, the program reads

```
10 DEF SEG&HF000
20 FOR X&HFFF5 TO &HFFFC
30 PRINT CHR$(PEEK(X));
40 NEXT
```

and for the Compaq Portable, the article lists the program as

```
10 DEF SEG&HF000
20 PRINT CHR$(PEEK(&HFFE6));
```

In both programs, line 10 should read instead as the following:

```
10 DEF SEG = &HF000
```

Running either program as listed by Mr. Aitken will result in a syntax error message in line 10.

Guillermo Hakim
New York, NY

Thank you for your corrections to the BIOS check listing for the two machines. Line 20 in the IBM PC check should be changed as well, to read

```
20 FOR X = &HFFF5 TO &HFFFC
```

Please also note some other corrections to this article's table of benchmark results (table 3, located across pages 84 and 85). First, under Measured Data, the overall average for the AUTOTEST random 8-sector read (0.90 width) actually should be 185.8

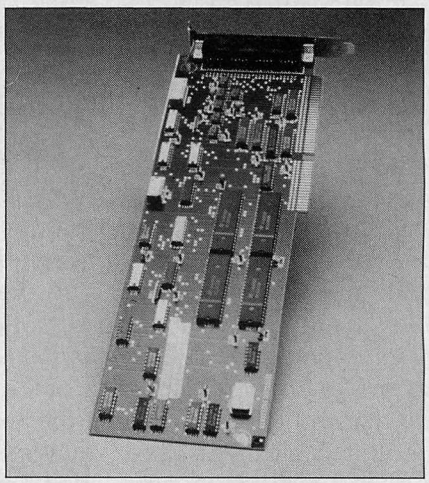
Under the next section, Percentage of Average Performance, several clarifications should be noted. First, the average of random tasks for the Mountain Computer card should say 83. Next, the unit measure for the ATDISK effective transfer rate should read milliseconds/KB (ms/KB)—designed so that a smaller number indicates a better performance (consistent with the remainder of the table). Also, to correct a computational error, the actual results for this test can be obtained by dividing the figures printed by 1.09.

Finally, the averages at the bottom of the table are computed as follows. The average for random tasks is obtained as an average of two items: (1) taking the average of random tasks in ATDISK, that is, the average seek time and the track-to-track seek time, and (2) the average of all eight random tests. The average of sequential tasks is obtained as an average of two items: (1) taking the average of sequential tasks in ATDISK, that is DOS file I/O and effective transfer rate, and (2) the average of sequential tests. The overall is an average of the average random and the average sequential results. Although the differences in the ATDISK effective transfer rate results (mentioned above) affect the sequential tasks average as well as the overall average, the conclusions that were drawn based on the results remain the same.

—Peter G. Aitken

ERRATA

The board that appeared in photo 13 (on page 62) of "RT PC: A Significant Departure" (Thomas V. Hoffmann, December 1986, p. 56) is actually the RT Monochrome Display Adapter, not the RT Multiport Communications Adapter (MCA). The MCA is pictured below.



In "Photo Plotting," the sidebar to "End-to-End Design" (Richard Angell, December 1986, p. 155), the first sentence of the second paragraph should say "With the introduction of computer-aided design (CAD), handtaping *is replaced by* pen plotting."

COMMENT AT WILL

All letters to the editor should be directed to Editor, *PC Tech Journal*, Suite 800, 10480 Little Patuxent Parkway, Columbia, MD 21044. Correspondence also can be submitted over MCI Mail to PCTECH.

Although *PC Tech Journal* cannot publish every letter received, every attempt is made to answer as many as possible. Please keep letters brief and to the point, and include name, mailing address, and telephone number; when a letter is lengthy, a diskette is appreciated.

PC Magazine
Product of the Year

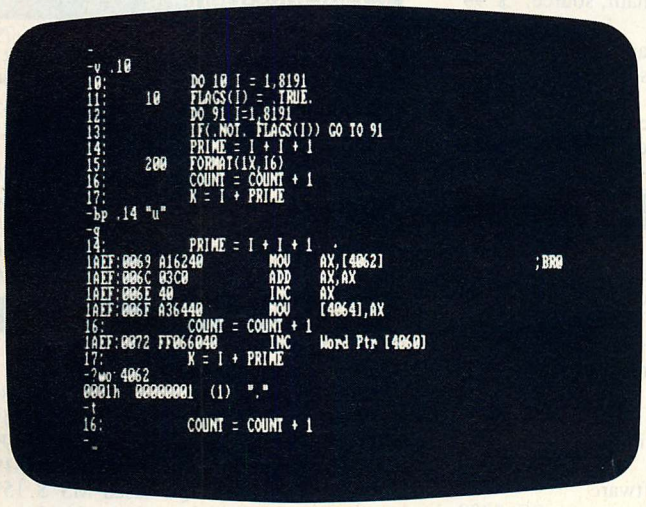
Macro Assembler

The quickest. Bar none.

Our Macro Assembler has long been the most complete package on the market. Now it's also the fastest. Three times faster than before. And faster than anyone else. Period.

Of course, it's still the most powerful assembler on the market. It supports the standard 8086/8087 opcodes. And the new 186/286/287 instruction set. So you can make the most of the new machines.

Debugging is quicker, too. Thanks to our interactive symbolic debugger, SYMDEB. Now you can refer to variables and source code instead of getting lost in hex dumps. And this debugger also works with Microsoft languages like C, FORTRAN and Pascal. So now you can set breakpoints and trace execution—using source code for reference.



Cut your development time dramatically. Microsoft Macro Assembler's Symbolic Debug utility lets you debug your Macro Assembler programs, or debug your Microsoft C, FORTRAN or Pascal programs using your original source code or the resulting disassembly. For example, you can set breakpoints on line numbers and observe the contents of variables or expressions.

SYMDEB is just part of our complete set of utilities. Tools that make programming as fast as it should be. There are the linker and library managers you'd expect. Plus a new version of MAKE, our maintenance utility, with improvements like macro expansions and inference rules.

We've also revised the manuals. Our new Macro Assembler has a lot to offer, so we added more examples. Now our manuals are not only thorough, they're clearer than ever before.

For quick development and assembly, the choice is obvious. Microsoft. There's nobody faster.

Microsoft® Macro Assembler Version 4.0 for MS-DOS®

Macro Assembler
- Fastest macro assembler for MS-DOS computers.
- Supports the 8086/8087/8088 and the 186/286/287.
- Define macros.
- Conditional assembly.
- Optional case sensitivity for symbols.
- 100% upward compatibility from earlier versions of both the Microsoft and IBM® Macro Assemblers.

Interactive Symbolic Debug Utility
- Source level debugger for programs written in Microsoft Macro Assembler, C Compiler, FORTRAN, and Pascal.
- Screen swapping helps debug highly visual applications.
- Set breakpoints on line numbers and symbols.
- Single step to follow program execution.
- Disassemble object code.
- Display and modify values.
- Full I/O redirection.

Program Maintenance Utility
- Rebuilds your applications after your source files have changed.
- Similar to UNIX™ MAKE utility.
- Supports macro definitions and inference rules.

Library Manager
- Create, organize and maintain your object module libraries created with Microsoft languages.
- Set page size from 16 to 32678, to create compact and granular libraries.

Object Code Linker
- Simple overlaying linker combines relocatable object modules created using Microsoft languages into a single program.
- Load Map generation.
- Specify from 1 to 1024 segments.

Cross-Reference Utility
- Creates a cross-reference listing of the definitions and locations of all symbols used in an assembly language program, which makes debugging programs easier.

Microsoft EXE File Compression Utility
- Packs EXE files for smaller size on disk and faster loading at execution time.

Microsoft EXE File Header Utility
- Display and modify EXE file header, allowing you to tune the stack size and initial memory allocation.

For the name of your nearest Microsoft dealer call (800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638.

Microsoft

The High Performance Software™

CIRCLE NO. 237 ON READER SERVICE CARD

Microsoft and MS-DOS are registered trademarks and The High Performance Software is a trademark of Microsoft Corporation. IBM is a registered trademark of International Business Machines Corporation. UNIX is a trademark of AT&T Bell Laboratories.

THE PROGRAMMER'S SHOP

helps save time, money and cut frustrations. Compare, evaluate, and find products.

RECENT DISCOVERY

The Documentor - for dBASE program flow chart, tree diagrams, .DBF documentation, variable/field concordance, hierarchy charts. Macros, searches, configure options. MS $ 295

AI-Expert System Dev't

Arity System-incorporate w/C.	MS	$ 259
Experteach-Improved, samples	PC	$ 349
EXSYS	PC	$ 339
Insight 2+ - dB2, language	MS	$ 389
Texas Instruments:		
PC Easy	PC	$ 439
Personal Consultant Plus	PC	$2599

AI-Lisp

Microsoft MuLisp 85	MS	$ 179
PC Scheme LISP - by TI. SCHEME has simple, "orthogonal" syntax.	PC	$ 85
TransLISP - Good for learning	MS	$ 85
TransLISP PLUS -		
Optional Unlimited Runtime		$ 150
PLUS for MSDOS		$ 179

Others: IQ LISP ($155), IQC LISP ($269)

AI-Prolog

APT - Active Prolog Tutor - build applications interactively	PC	$ 65
ARITY Standard - full, 4 Meg		
Interpreter - debug, C, ASM	PC	$ 319
COMPILER/Interpreter-EXE	PC	$ 699
With Exp Sys, Screen - KIT	PC	$1129
LPA MacProlog - Complete incremental compiler and an interpreter	MAC	$ 295
LPA MicroProlog - intro	MS	$ 85
LPA MicroProlog Prof. - full memory	MS	$ 339
Prolog-86 - Learn Fast	MS	$ 89
Prolog-86 Plus - Develop	MS	$ 229
TURBO PROLOG by Borland	PC	$ 69

Editors for Programming

BRIEF Programmer's Editor	PC	Call
EMACS by UniPress - Source: $929		$ 299
Epsilon - like EMACS	PC	$ 155
Kedit - like XEDIT	PC	$ 105
Lattice Screen Editor	MS	$ 109
PC/VI - by Custom Software	MS	$ 109
Personal REXX -	PC	$ 109
PMATE - power, multitask	PC	$ 109
SPF/PC - fast, virtual memory	PC	$ 139
XTC - multitasking	PC	$ 79

FEATURES

PolyBoost - Run 2 to 10 times faster with software accelerator. Speeds disk access, screen display, keyboard input. PC $ 69

C Scape - capture Dan Bricklin's, 1-2-3. Turbo screens & more, convert to C. Plus full screen generation package - tiled, pop-up windows with scrolling, validation. Source PC $179

Note: All prices subject to change without notice. Mention this ad. Some prices are specials. Ask about COD and POs. Formats: 3" laptop now available, plus 200 others. UPS surface shipping add $3/item.

National Accounts

MIS, Engineering, and Research departments get special FREE consulting, product comparisons, reports, newsletters. Compare approaches to COBOL, C, AI. PURCHASING AGENTS - get help and special service finding products, negotiating license agreements, with billing and more. Call 800-446-1185.

Our Services:

- Programmer's Referral List
- Compare Products
- Help find a Publisher
- Evaluation Literature FREE
- BBS - 7 PM to 7 AM 617-826-4086
- Dealers Inquire
- Newsletter
- Rush Order
- Over 700 products
- *National Accounts Center*

C Support-Systems

Basic-C Library by C Source	MS	$139
C Sharp - well supported.	PC	$600
C ToolSet - DIFF, xref, source	MS	$ 95
The HAMMER by OES Systems	PC	$149
Lattice Text Utilities	PC	$ 89
Multi-C - multitasking	PC	$149
PC LINT-checker. Amiga $89,	MS	$107
SECURITY LIB - add encrypt to MSC. C86 programs. Source $229	PC	$115
Quickshell - script compiler	PC	$349

Fortran & Supporting

50: More FORTRAN - math, source	$ 99
ACS Time Series	$419
Forlib+ by Alpha - graph, comm.	$ 59
MACFortran by Microsoft	$229
MS Fortran link to C	**$209**
No Limit - Fortran Scientific	$115
R/M Fortran - enhanced "IBM Ftn"	$389
Scientific Subroutines - Matrix	$139

MultiLanguage Support

BTRIEVE ISAM	MS	$199
BTRIEVE/N - multiuser	MS	$469
CODESIFTER - Profiler	MS	$ 99
Dan Bricklin's Demo Program	PC	$ 65
HALO Graphics - 115+ device interfaces, rich, printer. Specify language interface	PC	$209
Informix - by RDS	PC	$639
Informix 4GL - application builder	PC	$799
Microsoft Windows Software Development Kit	PC	$329
Opt Tech Sort - sort, merge	MS	$119
PANEL - Xenix $539	MS	$229
Pfinish Performance Analyzer	MS	$249
PLINK-86 - a program-independent overlay linker to 32 levels.	MS	$249
PLINK-86 PLUS - incremental	MS	$369
PolyLibrarian by Polytron	MS	$ 79
PVCS Version Control	MS	$329
Screen Sculptor - slick, thorough	PC	$ 99
ZAP Communications - VT 100	PC	$ 89

C Libraries-Communications

Asynch by Blaise	PC	$135
Greenleaf Comm Lib.	PC	$149
Multi-Comm - add multitasking, use w/Multi-C	PC	$149
Software Horizons pack 3	PC	$119

RECENT DISCOVERY

r-tree - report generation for ctree. Multiple handling, fixed or variable length. Many built-in functions like Boolean, computational functions, string, date handling, numeric to string conversion. Layout control. Source in C. PC $249

C Language-Compilers

AZTEC C86 - Commercial	PC	$ 499
C86 by CI - 8087, reliable	MS	$ 299
Datalight C - fast compile, good code, 4 models, Lattice compatible, Lib source. Dev'rs Kit	PC	$ 77
HOT C - new, intriguing	PC	$ 85
Lattice C - from Lattice	MS	$ 289
Mark Williams - w/debugger	MS	$ 369
Microsoft C 4.0 - CodeView	MS	$ **279**
Wizard C - full, fast.	MS	$ 359

C Language-Interpreters

C-terp by Gimpel - full K & R	MS	$ 229
C Trainer by Catalytix	PC	$ 89
INSTANT C - Source debug, Edit to Run-3 seconds, .OBJs	MS	$ 379
Interactive C - interpreter, editor	PC	$ 225
Introducing C - learn C quickly	PC	$ 105
Run/C Professional -	MS	$ 179
Run/C Lite	MS	$ 97

C Libraries-General

Blackstar C Function Library	PC	$ 79
C Essentials by Essential	PC	$ 83
C Food by Lattice-ask for source	MS	$ 99
C Scientific Subroutines-Peerless	MS	$ 135
C Tools Plus (1&2) - Blaise	PC	$ 135
C Utilities by Essential - Comprehensive screen graphics, strings. Source.	PC	$ 137
C Worthy Library	MS	$ 269
Entelekon C Function Library	PC	$ 119
Greenleaf Functions-portable, ASM		$ 139
PforCe by Phoenix - objects	PC	$ 229

C Libraries-Files

FILES: C Index by Trio - full B + Tree, vary length field, multi compiler		
/File is object only	MS	$ 89
/Plus is full source	MS	$ 319
CBTREE-Source, no royalties	MS	$ 99
CTree by Faircom-no royalties	MS	$ 319
dbQUERY-ad Loc, SQL-based	MS	$ 159
dbVISTA - full indexing, plus optional record types, pointers, Network.		
Object only - MS C, LAT,	C86	$ 155
Source - Single user	MS	$ 425
Source - **Multiuser**	MS	$ 845
dBASE Tools for C	PC	$ 65
dbc Isam by Lattice	MS	$ 179
dBx - translator	MS	$ 319
w/source to library	MS	$ 499

FEATURE

Uniware Cross Development Tools - include 68000 C compiler. Development Package with compiler, assembler, link editor, and utilities. 17 cross assemblers for Intel, TI, Motorola, Zilog, etc. - relocatable, macros. MS Call

We support MSDOS (not just compatibles), PCDOS, Xenix-86, CPM-80, Macintosh, Atari ST, and Amiga.

CIRCLE NO. 162 ON READER SERVICE CARD

THE PROGRAMMER'S SHOP

provides complete information, advice, guarantees and every product for Microcomputer Programming.

Special Features

COBOL & C Screen Management

Finally . . . Easy Screen Handling for COBOL Saves Time, Adds Flexibility: Screenplay

Menus, Help, and Data Entry Screens can be created ("painted") interactively. When you are satisfied Screenplay performs all screen handling for you in any of more than 9 compilers.

Save valuable time by avoiding the tedious, time consuming process of writing screen handling source code. Screenplay's easy-to-use panel painter allows you to create I/O panels, pop-up windows or menu panels.

True screen handling flexibility is yours. Override default panel settings to design practically any type of screen imaginable — even change just about any panel characteristic at runtime.

Prototype your draft screens before you write a single line of COBOL source code. In addition, you can link Screenplay by interrupt or directly to your application. And if your compiler doesn't allow a direct link, use the dynamic load option for linking Screenplay to your application.

You can assign practically any keyboard key to serve as a specific cursor function and define exactly which keys will return control to your application. Entirely reconfigure the keyboard for your program.

Screenplay's panel painting process is a one step approach. There's no need to go through a separate process to establish fields on your I/O panel. What's more, you can use any ASCII character in your Screenplay panels. You also have full control over character attributes such as foreground and background color, intensity, and blinking.

Panels are stored in a compressed ASCII file save memory and disk space. Copy panels across and within files, rename, delete, test and print panel details. Even print an image of the panel for your documentation. No royalties. Not copy-protected. Supports IBM, MS, Realia, Ryan-McFarland (including 8X) COBOL.

201-895-4724 **PCDOS List: $175 Ours: $155**

Multitasking Technology

Multitasking, Windowing for C, Turbo Pascal, or dBASE or . . . in only 12K!

SYNERGY Development Toolkit

The highly efficient design of Synergy by Matrix gives you the benefits of powerful graphics, windows, pull-down menus, dialog boxes, sophisticated text and icon management, math support, multitasking, and SPEED, all for an incredibly small **12K RAM** requirement.

The Synergy Runtime provides character and graphics support for menus, windows, dialog boxes, and more, so you can write programs that work in either mode, with very reasonable, low runtime fees.

Functions include: window management with capabilities like tile and overlap, variable size and placement, process management, to support multitasking and sub-process generation, menus, dialog, and icon management, graphics, text (including a variety of fonts and sizes), and console management.

The Synergy Development Toolkit is a collection of sophisticated tools designed for software developers writing new applications using Synergy, or modifying existing applications to take full advantage of Synergy. Tools include: graphics resource editor for creating and modifying icons and text fonts, graphics resource compiler to construct and manage resource data files, font compiler and manager, debugging tools, sample library, and more.

Supports IBM or Microsoft Macro Assemblers, Turbo, IBM, and Microsoft Pascal, IBM and Microsoft BASIC, Lattice and Microsoft C, and dBASE II and III. CGA, EGA, and Hercules monochrome graphics support.

MATRIX **617-567-0037** **PCDOS List: $395 Ours: $375**

Expert System Shell & Language

Add AI Power to Existing Applications

Insight 2+

"Insight 2+ is an awesome system . . . this may well be the best expert system generator on the market." — ONLINE

Choose a knowledge engineering tool that makes your current database solutions work smarter with a built-in PASCAL-like knowledge engineering grammar that minimizes your training time. Insight 2+ works with your installed software and database and has intrinsic dBASE II and III operators, plus Level Five's powerful, flexible Production Rule Language.

Insight 2+ is a highly connective, integrated expert systems shell that adapts to your needs. Its detailed interactive reporting system lets you see *inside* the reasoning process of the knowledge base. You can see reports on the lines of reasoning, known and unknown facts, alternate rule pathways to a shared conclusion, or knowledge trees, showing all possible goals, conclusions, and paths of reasoning hierarchically — and more. You can even step forward and back in a rule chain from beginning to end.

Create "what-if" scenarios by saving, editing, and re-using knowledge base contexts, or customized reporting to track end-user sessions.

Insight 2+ performs backward and forward chaining inference, supported by object attribution and goal outlining, outside program activation, and full parameter passing capabilities.

Insight 2+ supports very large knowledge bases which can be linked — the number limited only by disk space. Full scientific and math capabilities include logarithmic and trigonometric functions and floating point notation; ideal for engineering and scientific applications.

305-729-9046 **PCDOS List: $485 Ours: $389**

Screen Management

Get a Cleaner, Faster, User Interface with

The Screen Ace Form Master

Coded in assembler to give you the power for creative design, and the speed not available in other packages.

Form Master supports all 256 color and monochrome attributes. Paint screens with the screenbuilder (allows changes without recompiling!) and save to a DOS file, define screens within your program at runtime, or combine these techniques, to modify screens on the fly.

400+ lines and 2000+ fields per screen; the number of screens limited only by available memory.

Permits flexible function key definition, toggling field colors and attributes at runtime. Switch between screens without losing data.

DESQview, TopView, MS Windows compatible, Lattice, MS C (all models), APL*Plus/PC, Assembler. Call for a $3 demo with tutorial.

PCDOS List: $195 Ours: $179
818-989-5329

O/S ADDON

NOW: Full Shell Programming with MSDOS:

PolyShell with **UNIX-Like Utilities**

PolyShell gives you a UNIX-compatible command environment PLUS a full set of UNIX-like utilities — yet DOS commands and batch files are always immediately accessible.

Alias and Command Macro Substitution save keystrokes, let you access and customize DOS internal commands. With History Substitution, edit, re-execute previous commands or portion of command.

'C-Shell' and Bourne Shell compatible I/O redirection, very flexible filename expansion capabilities.

Includes 30 powerful utilities from the UNIX world (like grep, diff, sed, cut, paste). Shell and utilities can be loaded separately. Requires only 30K RAM total. PolyShell is NOT UNIX; no UNIX license needed.

PCDOS List: $149 Ours: $129
503-645-1150

Call for a catalog, literature, advice and service you can trust

VISA **HOURS** MasterCard
8:30 AM - 8:00 PM EST.

800-421-8006

THE PROGRAMMER'S SHOP™
5 Pond Park Road, Hingham, MA 02043
Mass.: 800-442-8070 or 617-826-7531 1/87

"The scope and detail of services you provide are exemplary — it's obvious you have given a lot of thought to what information people need . . . For someone like myself, critical appraisals of software and comprehensive collections of offerings such as you have are really useful."

A. Bruce Cyr
Foundation of American College of Health Care Administrators

CIRCLE NO. 151 ON READER SERVICE CARD

The fastest, tightest code.



(Though the same can hardly be said of the name.)

We have to tell you, we had a hard time getting the name down this short.

Because Microsoft's new FORTRAN Compiler actually has a far longer list of features.

It uses the same optimizer and code generator technology that made our C Compiler the industry leader.

And we've added special loop optimizations that give you the smallest, fastest FORTRAN code a PC can handle.

> *"Now Microsoft's FORTRAN Optimizing Compiler generates such fast code that an IBM PC/XT approaches the speed of the VAX."*
>
> *Peter Osgood, MIT, Project Athena, Director of the Real Time Lab Project.*

This compiler has already passed the toughest test there is. It's been

Microsoft FORTRAN Optimizing Compiler Version 4.0.

- Full ANSI FORTRAN 77 Compiler with Fastest Executable Code for MS-DOS. NEW!
- Uses the Microsoft C optimizing technology, plus loop optimization. NEW!

Execution Speed (in Seconds)	Microsoft FORTRAN v. 4.0	Ryan-McFarland FORTRAN v. 2.11	IBM Professional FORTRAN v. 1.22
Sieve	7.97	9.33	38.51
Whetstone	53.82	58.67	79.04
Lookup	5.82	18.61	26.02

- Fully GSA certified for ANSI 77 compatibility with no errors at the highest level. NEW!
- Numerous IBM VS and DEC VAX extensions. NEW!
- Microsoft CodeView: Window-oriented source-level debugger. NEW!
  - —Debug using your original source code, the resulting disassembly or both intermingled.
  - —Watch and change the values of your local and COMMON variables as you debug.
  - —Set conditional breakpoints on variables, expressions or memory; trace and single step.
  - —Watch and change registers and flags as you execute.
  - —Easily debug graphics oriented programs since program output is kept separate from debugger output.

GSA-certified as Full ANSI FORTRAN 77, and 100% error-free.

"The Microsoft FORTRAN Optimizing Compiler let us port the 200,000 line Boeing Mathematical Library (BCSLIB) with virtually no changes. This ANSI FORTRAN 77 code was ported directly from Cray, CDC, DEC, IBM and other mainframes and workstations."

Ivor Philips, Boeing Computer Services, Program Manager Mathematical Software Libraries.

Compiler V. 4.0 with CodeView™

We've also included the same advanced intrinsic math functions found on VAX® and IBM® VS systems. Add improvements like our new HUGE memory model, and porting the biggest mainframe programs has never been easier.

Among the many additions we've made to our package is our exclusive CodeView™ windowing debugger. It lets you trace through programs at any level you want, from source code to assembly language.

You can open windows, and watch both variables (local and COMMON) and CPU registers change. You can set conditional breakpoints using variables and expressions.

Debugging gets even easier with the compiler's advanced diagnostics. Detailed error messages are thoroughly explained and cross-referenced in our new manuals.

Documentation that has been completely revised and expanded with tons of examples.

If we're talking your language, use one of the numbers below for more details about the Microsoft® ANSI FORTRAN 77 Optimizing Compiler Version 4.0 with CodeView, and the name of your nearest dealer.

(Even if the call's toll-free, it may be a good idea to refer to it as "FORTRAN 4" for short.)

Microsoft® FORTRAN

The High Performance Software.

Call (800) 426-9400. In Washington State or Alaska, (206) 882-8088. In Canada, (416) 673-7638.

Microsoft and MS-DOS are registered trademarks and CodeView is a trademark of Microsoft Corporation. IBM is a registered trademark of International Business Machines Corporation. VAX is a registered trademark of Digital Equipment Corporation.

- Medium, Large and Huge Memory Model Libraries. NEW!
- Mix models with NEAR, FAR and new HUGE pointers.
- Common blocks and arrays greater than 64K.
- Choose from three math libraries and generate in-line 8087/80287 instructions or floating point calls:
  - —floating point emulator (utilizes 8087/80287 if installed)
  - —8087/80287 coprocessor support
  - —alternate math package—extra speed without an 8087/80287
- Link your FORTRAN routines with Microsoft C (v. 4.0 or higher), Microsoft Pascal (v. 3.3 or higher) or Microsoft Macro Assembler.
- Largest number of 3rd party support libraries available.
- Provides more detailed diagnostic error messages (almost twice as many as competitors) and extensive documentation with non-ANSI 77 features highlighted. NEW!
- Proven reliability—tested with over 2.5 million lines of code compiled and executed.
- MS-DOS® network support with file / record locking and sharing.
- Microsoft Program Maintenance Utility rebuilds your applications after your source files have changed. NEW!
- Other utilities including faster overlay linker (links over 1Mbyte object code), library manager, EXE file compression utility, EXE file header utility, MS-DOS environment setting utility and setup utility.

CIRCLE NO. 205 ON READER SERVICE CARD

The fastest C

Your search for execution speed is over. The new Microsoft® C Compiler Version 4.0 is here. With blazing performance. We've added common sub-expression elimination to our optimizer that produces code that rips through the benchmarks faster than ever before.

"...the Microsoft performance in the benchmarks for program execution is the best of the lot overall." —William Hunt, *PC Tech Journal*, January, 1986.*

But speed isn't the only edge you get with Microsoft C. Other advantages include a variety of memory models like our new HUGE model that breaks the 64K limit on single data items. Plus our NEAR, FAR and HUGE pointers, which provide you greater flexibility. All this allows you to fine tune your program to be as small and fast as possible.

"Excellent execution times, the fastest register sieve, and the best documentation in this review ...Microsoft Corporation has produced a tremendously useful compiler." —Christopher Skelly, *Computer Language*, February, 1986.

No more debugging hassles. Introducing CodeView. Free.

Now, for a limited time, we'll give you an unprecedented programming tool when you buy Microsoft C, free. New Microsoft CodeView™ offers the most powerful tool yet in the war on C bugs. Forget the hex dumps. Now you can view and work with programs at any level you want. Use the program source, the disassembled object code, or



Microsoft C Compiler Version 4.00

Microsoft C Compiler

- Produces fast executables and optimized code including elimination of common sub-expressions. NEW!
- Implements register variables.
- Small, Medium and Large Memory model libraries.
- Compact and HUGE memory model libraries. NEW!
- Can mix models with NEAR, FAR and the new HUGE pointers.
- Transport source and object code between MS-DOS® and XENIX® operating systems.
- Library routines implement most of UNIX™ System V C library.
- Start-up source code to help create ROMable code. NEW!
- Full proposed ANSI C library support (except clock). NEW!
- Large number of third party support libraries available.
- Choose from three math libraries and generate in-line 8087/80287 instructions or floating point calls:
  - floating point emulator (utilizes 8087/80287 if installed).
  - 8087/80287 coprocessor support.
  - alternate math package – extra speed without an 8087/80287.
- Link your C routines with Microsoft FORTRAN (version 3.3 or higher), Microsoft Pascal (version 3.3 or higher) or Microsoft Macro Assembler.
- Microsoft Windows support and MS-DOS 3.1 networking support.
- Supports MS-DOS pathnames and input/output redirection.

Microsoft Program Maintenance Utility. NEW!

- Rebuilds your applications after your source files have changed.
- Supports macro definitions and inference rules.

Other Utilities

- Library Manager.
- Object Code Linker.
- EXE File Compression Utility.
- EXE File Header Utility.

C Benchmarks

In seconds

	Microsoft C 4.0	Lattice C 3.0	Computer Innovation C 2.3	Aztec C86 3.2	Wizard C 3.0
Sieve of Eratosthenes (register)	82.9	151.4	172.3	88.0	91.9
Copy Block	86.9	231.7	199.0	123.8	189.5

Run on an IBM PC XT with 512K memory

Microsoft CodeView
Window-oriented source-level debugger. NEW!

- Watch the values of your local and global variables and expressions as you debug.
- Set conditional breakpoints on variables, expressions or memory; trace and single step.
- Watch CPU registers and flags as you execute.
- Effectively uses up to four windows.
- Debug using your original source code, the resulting disassembly or both intermingled.
- Use drop-down menus to execute CodeView commands.
- Access the on-line help to lead you through CodeView's options and settings.
- Easily debug graphics-oriented programs since program output is kept separate from debugger output.
- Keyboard or optional mouse support.
- Enter in familiar SYMDEB or DEBUG commands.

*Reprinted from *PC Tech Journal*, January 1986, copyright 1986, Ziff-Davis Publishing.

you've ever seen.

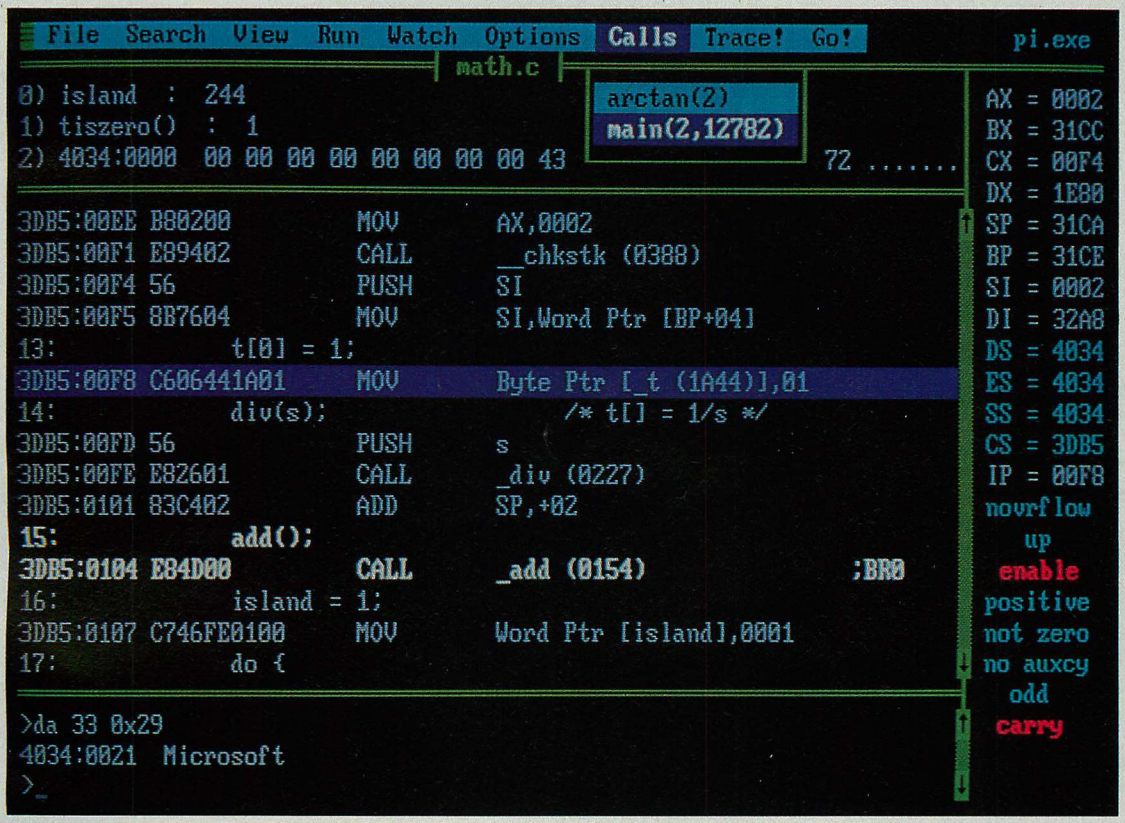
both at the same time. Open a window to view CPU registers and flags. Watch local and global variables as well. All while your program is running.

CodeView gives you complete control. Trace execution a line at a time—using source or assembly code. Or set conditional breakpoints on variables, memory or expressions. CodeView supports the familiar SYMDEB command syntax, as you'd expect. Commands are also available through drop-down menus. Combine the new window-oriented interface with our on-line help and debugging has never been easier. Or quicker.

Take the $5 CodeView tour.

You may find it hard to believe our debugger can do all we've claimed. So we're offering test drives. Five bucks will put you behind the wheel of a Microsoft C demo disk with CodeView.† See for yourself how fast debugging can get.

For more information about the CodeView demo disk, the new Microsoft C Compiler, a list of third party library support or the name of your nearest Microsoft dealer, call (800) 426-9400. In Washington State and Alaska, (206) 882-8088. In Canada call (416) 673-7638.



Microsoft® C Compiler

The High Performance Software

Microsoft, MS-DOS and XENIX are registered trademarks and CodeView is a trademark of Microsoft Corporation. UNIX is a trademark of AT&T Bell Laboratories. IBM is a registered trademark of International Business Machines Corporation. †Offer expires 12/31/86.

CIRCLE NO. 140 ON READER SERVICE CARD

Finally, a language worth

For years BASIC has been everyone's first language. And for almost as long, they've been tempted by other languages. Lured by promises of more speed, more power.

We have a solution. A new language that's a substantial improvement over BASICA. Faster. More structured. Finally, a compelling reason to leave BASIC.

Introducing Microsoft's QuickBASIC Compiler, Version 2.0.

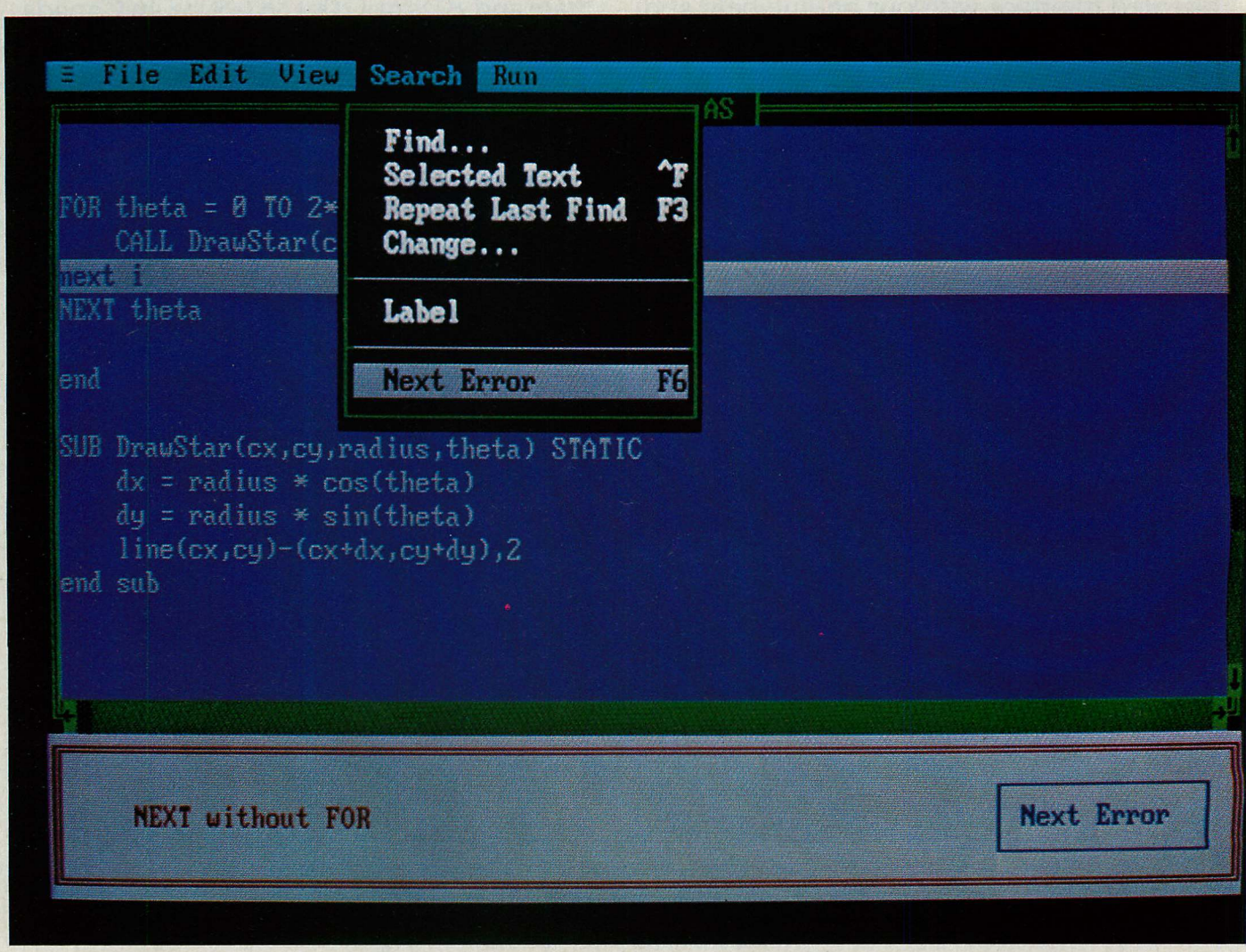
At last, you can have the latest programming techniques, combined with the solid foundation of BASIC. Our new compiler is as compatible with BASICA as you can get. At the same time it offers the extra speed and power you've been looking for.

Run faster with compiled code.

If there's one thing you've asked for, it's speed. And Microsoft® QuickBASIC simply blazes. Old BASICA programs will run up to ten times quicker once they've been compiled. Sometimes even faster.

Everything you need. Built-in.

Making programs run faster is only part of the story, though. The new Microsoft QuickBASIC Compiler includes a full-screen editor, built-in. So now you can make the jump from writing to RUNning in no time flat. Edit your program, compile it, and run it. Faster than any other BASIC compiler around. All without leaving our on-line help and prompts.



Microsoft is a registered trademark and The High Performance Software is a trademark of Microsoft Corporation. IBM is a registered trademark of International Business Machines Corporation.

leaving BASIC for.

On the rare chance your program doesn't run 100% the first time out, we've got another surprise for you. The Microsoft QuickBASIC debugger. Our full-screen tracing lets you debug your programs while watching the source code execute. A line at a time, or with breakpoints. As easy as can be.

Our compiler is also smart enough to save you time. First, by finding any errors in one pass. Second, by putting your editor's cursor on the problem. Automatically. So you don't have to get lost in a maze of error codes and line-numbers.

The BASIC virtues. And more.

Speaking of line numbers, let's not. Because line numbers are strictly optional. And Microsoft QuickBASIC lets you use alphanumeric labels as well. Now you can GOTO *ErrorCheck* instead of line number 6815.

Or you could stop using GOTOs altogether. There are a variety of options that could make the GOTO an endangered species. Features like multi-line IF-THEN blocks. And named subprograms. Now your BASIC programs can be as structured and organized as you want.

We've only just begun to talk about the virtues of Microsoft QuickBASIC. There are dozens of enhancements to your favorite language. Things like larger arrays. Local and global variables. Reusable modules that let you create libraries of your most often-used routines. All explained in a revised manual that includes a complete language reference.

Making your quick escape.

If all these features follow your BASIC instincts, then zip on down to your nearest Microsoft dealer. That's where you'll discover the best surprise of all. The price. Only $99 for the best reason to leave BASIC.

For the name of your nearest Microsoft dealer, call (800) 426-9400. In Washington State and Alaska, (206) 882-8088. In Canada, call (416) 673-7638.

Microsoft QuickBASIC Compiler Version 2.0 for IBM® PC and Compatible Computers.

BASICA Compatibility
- Sound statements including SOUND and PLAY.
- Graphics statements including WINDOW, VIEW, DRAW, GET, PUT, LINE, CIRCLE, LOCATE and SCREEN.
- Support of EGA extended graphics modes. NEW!
- BASICA structures are supported including WHILE/WEND, IF/THEN/ELSE, FOR/NEXT, GOSUB/RETURN, and event handling.

Results of Sieve Benchmark	BASICA 3.1	Microsoft QuickBASIC 2.0
Seconds per iteration	78	0.52

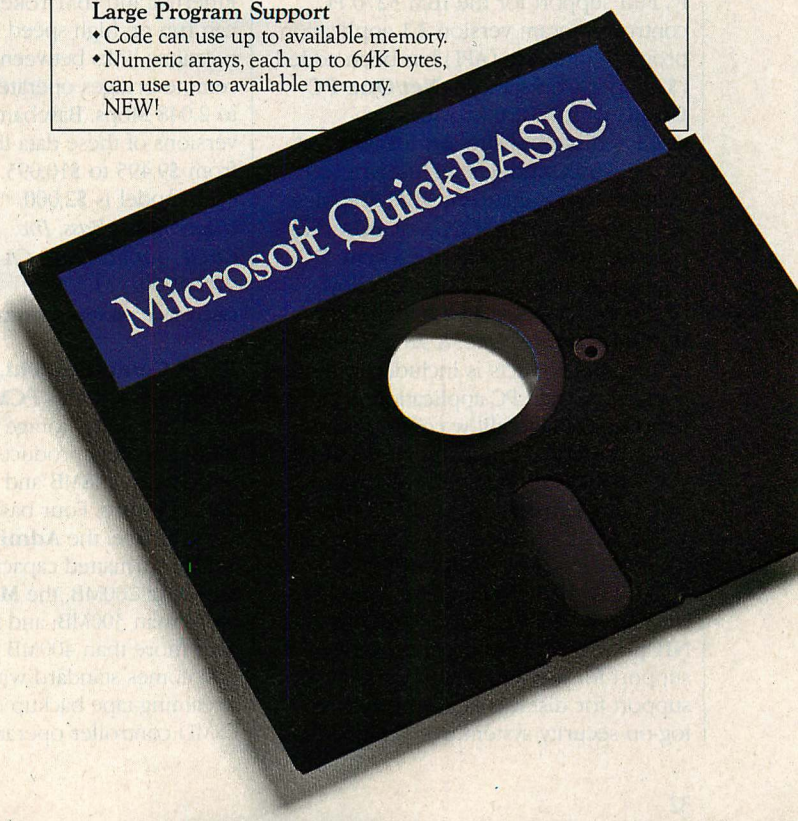
Complete Programming Environment
- Built-in Editor that places the cursor on found errors automatically. NEW!
- Compile entirely in memory at speeds up to 6000 lines per minute. NEW!
- Link routines once when starting a programming session and no need to link again when changing programs. NEW!
- Built-in debugger with single-step, animate, and trace modes. NEW!
- Create stand-alone programs.

Alphanumeric Labels
- Can be used to make your programs more readable. Line numbers are not required but are supported for BASICA compatibility.

Structured Programming Support
- Block IF/THEN/ELSE/END IF eliminates the need for GOTO statements. NEW!
- Subprograms can be called by name and passed parameters. Both local and global variables are supported.

Modular Programming Support
- Separate compilation allows you to create compiled BASIC libraries to use and re-use your programs.
- A library of routines to access DOS and BIOS interrupts is supplied. NEW!

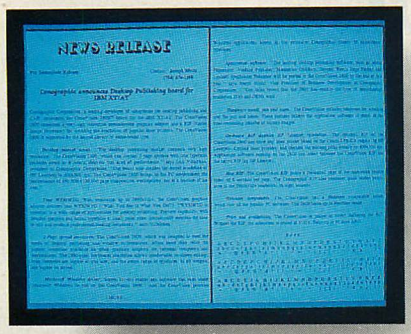
Large Program Support
- Code can use up to available memory.
- Numeric arrays, each up to 64K bytes, can use up to available memory. NEW!



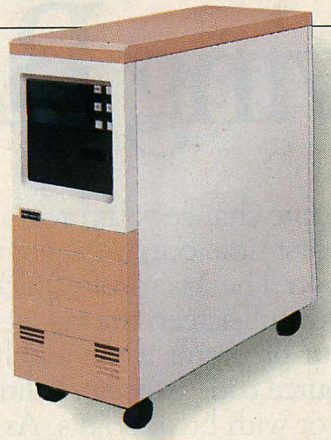
Microsoft® QuickBASIC
The High Performance Software™

CIRCLE NO. 121 ON READER SERVICE CARD

Hardware, software, and other developments for the IBM PC family



Screen produced with Conographic's ConoVision 2800 board



PCMS Director from Racet Computes, Ltd.

HARDWARE

The **Net/One Universal Workstation Series** is a group of hardware and software products from **Ungermann-Bass, Inc.**, designed to develop a local area network (LAN) for corporate use.

The **Net/One 3270 NIUpc** network interface unit (NIU) adapter card enables a networked PC to run micro-to-mainframe applications that are written to the IRMA (Digital Communications Associates, Inc.) and IBM 3278/79 Emulation Adapter hardware (Ethernet baseband version, $1,495; broadband, 2,145). **Net/One 3270 PC** software enables a PC/XT or PC/AT to function as a full-featured, networked IBM terminal. Net/One 3270 PC supports as many as four 3270 terminal and printer sessions, two asynchronous terminal sessions, and one DOS application simultaneously. Full support for the IBM 3270 PC control program version 2.1 application program interface (API) has been added ($295 per workstation). **Net/One PC Graphics** software provides programmed symbol support for the Net/One 3270 PC and lets the user run as many as four concurrent IBM graphics sessions with an IBM CGA or EGA ($295 per workstation). **Net/One Async PC** is communications software that allows PCs to exchange data with asynchronous resources on Net/One. A set of terminal emulation products is included with the Net/One Asynch PC application program interface (API) to allow communication with a variety of minicomputers ($1,200 per server when bundled with Microstuf, Inc.'s Crosstalk XVI; $1,600 per server when bundled with Crystal Point, Inc.'s P_CTERM—see SOFTWARE).

Working in conjunction with the Net/One PC software, the **Net/One NIUpc** adapter card provides server support for as many as 96 users, server support for diskless PCs, an extensive log-on security system that is embedded in network protocol, and a pop-up window utility that allows connections to Net/One resources to be made or deleted and network print queues to be viewed from within the PC application (Ethernet baseband version, $1,095; broadband, $1,745).

Ungermann-Bass also has introduced high-speed network bridges capable of linking local or geographically remote IBM-compatible token-ring LANs to an Ethernet network. Available in local and remote models, **Net/One Data Link Bridges** provide protocol-independent routing of data packets among multiple types of LANs. Bridges allow multiple communications protocols, such as Xerox Network System (XNS), Transport Control Protocol/Internet Protocol (TCP/IP), International Standards Organization (ISO), and Digital Equipment Corporation's DECnet to operate over the bridge. Both the 3Com Ethernet and IBM Token-Ring networks can run on high-speed T-1 telecommunications links between distant sites. All remote bridges operate from 4.8 Kbps to 2.048 Mbps. Baseband and broadband versions of these data link bridges range from $9,495 to $10,095. Software for each model is $2,000.

Ungermann-Bass, Inc., 3900 Freedom Circle, Santa Clara, CA 95052-8030; 408/496-0111

CIRCLE 301 ON READER SERVICE CARD

Racet Computes, Ltd. has announced an expansion of its **PCMS** (personal computer mass-storage) line of subsystems. The new products offer capacities of between 148MB and 870MB in a single enclosure. Four basic configurations are available: the **Administrator**, with 148MB formatted capacity; the **Supervisor**, with 200MB; the **Manager**, with more than 300MB; and the **Director**, with more than 400MB. Each configuration comes standard with 125MB streaming-tape backup capability. The ESMD controller operates at 2.4MB per second and has .5MB of cache memory that is expandable to 4.5MB. The average access time is 18 milliseconds (ms) for the Supervisor, Manager, and Director, and 25 ms for the Administrator. Prices range from $7,950 to $39,900; add-on 4MB cache, $3,750.

Racet Computes, Ltd., 1855 W. Katella, Suite 255, Orange, CA 92667; 714/997-4950

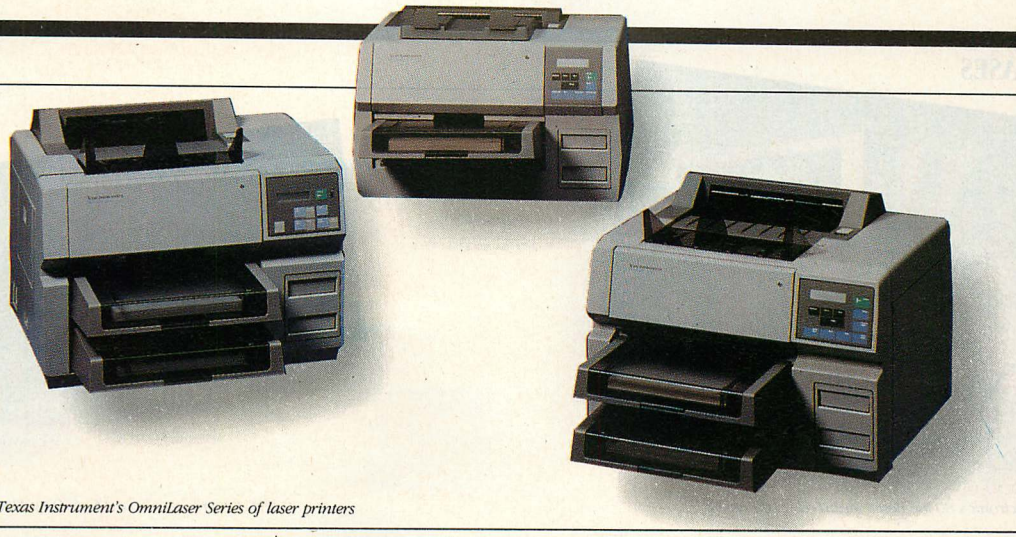
CIRCLE 310 ON READER SERVICE CARD

ConoVision 2800 is a desktop publishing board from **Conographic Corporation**. The board combines a high-resolution monochrome graphics adapter and a raster image processor (RIP) for doubling the resolution of popular laser printers. The ConoVision 2800 can display two-page spreads with true typefaces readable to as small as six points; it can double the output of Hewlett-Packard LaserJets to 600-by-300 dots per inch (dpi) and has a resolution up to 2,880-by-1,024 pixels. Screen drivers enable any software running under Microsoft Windows to run under ConoVision 2800. Hardware for scrolling, panning, and zooming is included. The board has a Hercules-compatible mode and drives 19-inch professional (100 MHz) and 15-inch high-performance (50 MHz) monitors. ConoVision 2800 with RIP, $1,985; without RIP, $1,325.

Conographic Corporation, 17841 Fitch, Irvine, CA 92714; 714/474-1188

CIRCLE 312 ON READER SERVICE CARD

Number Nine Computer Corporation introduces its **Pepper Graphics System** family with the **PEPPER PRO1280**. The PRO1280 combines the Number Nine Intelligent Operating System (a library of on-board, device-independent graphics, video control, and advanced memory-management functions) and the Memory Windows architecture, which allows mapping of IBM address space to the PRO1280 display or instruction memory, with Texas Instrument's TMS 34010



Texas Instrument's OmniLaser Series of laser printers

GSP graphics coprocessor. It combines noninterlaced 1,280-by-1,024-pixel resolution and 256 on-screen colors with IBM monochrome, CGA, and Professional Graphics Controller compatibility, in a single-board, single-monitor system. The board has 1.25MB of video RAM (expandable to 8MB), 128KB of instruction RAM, 128KB of graphics firmware, and either a 4,096- or a 16-million-color palette. The PRO1280 features a 16-by-38-pixel system font, as well as high display resolution. $3,000.
Number Nine Computer Corporation, 725 Concord Avenue, Cambridge, MA 02138; 617/492-0999
CIRCLE 307 ON READER SERVICE CARD

From **Epson America, Inc.** comes the **Image Scanner Option Kit**, an accessory for Epson's EX-800, EX-1000, and LQ-2500 dot-matrix printers. The scanner reads and converts hard-copy images into bit-image data, then transmits the data through a serial interface to a host computer where they are stored. The scanner offers a resolution of 180-by-180 dots per inch (dpi) on the LQ-2500, and 144-by-144 dpi on the EX series. A page of graphics can be scanned or digitized at 27 inches per second (ips) on the LQ-2500, and 25 ips on the EX series. $299.95.
Epson America, Inc., Computer Products Division, 2780 Lomita Blvd., Torrance, CA 90505; 800/421-5426; in California, 213/539-9140
CIRCLE 303 ON READER SERVICE CARD

A family of laser page printers that is designed for shared resource environments, the **OmniLaser Series 2000**, has been announced by **Texas Instruments**. The **OmniLaser 2015** has a print speed of 15 pages per minute (ppm), a maximum-duty cycle of 25,000 pages per month, and a machine life of 1.5 million prints. The **OmniLaser 2115** uses the same 15-ppm print engine as the 2015; its PostScript controller with 3MB of RAM is built around the 32-bit Motorola 68000 chip. The **OmniLaser 2108** has an 8-ppm engine with a duty cycle of 10,000 pages per month, a 600,000-print machine life, and a 68000-based PostScript controller with 2MB of RAM. OmniLaser 2015, $5,995; 2115, $7,995; 2108, $5,995.
Texas Instruments, Data Systems Group, P.O. Box 809063, H-861, Dallas, TX 75380-9063; 800/527-3500
CIRCLE 304 ON READER SERVICE CARD

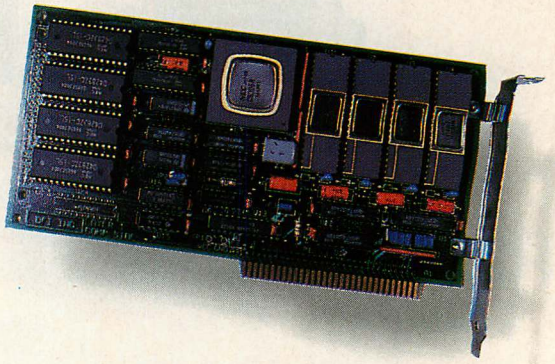
AST Research, Inc. has announced the **AST-5250/Gateway**, a single board with software that connects a LAN to an IBM System/3x machine. This gateway allows any LAN node to emulate 5250 terminal/printer sessions without an emulation card. The AST-5250/Gateway accommodates remote and local distribution systems in peer-to-peer or host communications link-ups. One microcomputer on the LAN is equipped with the AST 5251/11 Plus emulation card and a software package that distributes 5250 terminal/printer sessions to other nodes on the LAN. A maximum of three Gateways can coexist on the same LAN, and each Gateway can communicate simultaneously to a different host. Connections from the Gateway are made via standard twin-axial cable for local hosts or SNA/SDLC for remote hosts. $1,995.
AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714-4992; 714/863-1333
CIRCLE 313 ON READER SERVICE CARD

Three networking products have been introduced by **Racore Computer Products, Inc.** The **LANpac II**, a high-performance network system designed for Novell Advanced NetWare, can be connected in either a linear-bus or star configuration using coaxial or twisted-pair cable. It operates at selectable speeds of 4Mbit, 8Mbit, or 16Mbit per second. As many as 250 LANpac stations per cluster can be connected at a distance of up to 1,000 feet between stations. A **diskless workstation** incorporating optional LANpac circuitry for use in a PC/AT-compatible environment is available. An Intel 80286, operating at 10 or 12 MHz, provides AT performance. Available with LANpac directly integrated onto the main processor board, the workstation can accommodate as much as 2MB of memory. **LANpac 802.5** is a plug-compatible IBM Token-Ring adapter that can be incorporated into an existing IBM Token-Ring Network. LANpac 802.5 transfers data at the rate of 4Mbit per second using coaxial or twisted-pair cable. LANpac II, $295 per node; diskless workstation, $850 to $1,525; LANpac 802.5, $495 per node.
Racore Computer Products, Inc., 170 Knowles Drive, Los Gatos, CA 95030; 408/374-8290
CIRCLE 314 ON READER SERVICE CARD

Convergent Technologies has released several products for its network architecture, the **Cluster**. The **Series 386 NGEN** is an upgrade for current users of Convergent's NGEN workstations. Existing applications and peripherals are compatible with the new processor. **CTOS/VM**, a PC-compatible version of Convergent's distributed multitasking operating system, CTOS, supports multiple operating systems and their applications. It also provides support for 80286/80386 protected mode. The **ClusterCard**, which fits expansion slots on both the PC/AT and PC, and **ClusterShare** server software, provide network services, including electronic mail and resource sharing, to PCs over ClusterNet. **TeleCluster** piggybacks Cluster data on standard voice signals over existing in-house telephone wiring. The AT-compatible **Network PC** is designed to operate with or without local disk storage on the Convergent Cluster, 3 Com Ethernet, and other standard networks. Its built-in functions includes EGA video and up to 1MB of memory, in a desktop design. Retail prices are not applicable;



MultiTech Electronic's 80386-based MultiTech 1100 system



DF-1 coprocessor board from Data Flow Imaging, Inc.

all products are available in original equipment manufacturer (OEM) quantites and to value added retailers (VAR).
Convergent Technologies, 2700 N. First Street, P.O. Box 6685, San Jose, CA 95150-6685; 408/434-2848

CIRCLE 311 ON READER SERVICE CARD

A graphics board that is designed to run on the Sony Multiscan CPD 1302 monitor has been released by **QDP Computer Systems, Inc.** The **VIVA 1000/SCAN** with a resolution of 1,000-by-600 pixels and 16 simultaneous colors, displays a high-resolution, flicker-free image; it uses an NEC 7220A microprocessor. Features that are built into the board include hardware pan and zoom, and user-selectable cursors and colors, which are available via a single keystroke. The VIVA 1000/SCAN can be upgraded to QDP's 1000/16, which has a resolution of 1,024-by-1,024 pixels and drives 19-inch monitors. An available upgrade permits the display of 256 colors simultaneously from a palette of 16.8 million. Under $1,300.
QDP Computer Systems, Inc., 23632 Mercantile Road, Beachwood, OH 44122; 216/464-6600

CIRCLE 308 ON READER SERVICE CARD

Three low-cost interface boards that permit accurate monitoring and control of real world signals for the IBM PC, PC/XT, and PC/AT have been announced by **Real Time Devices, Inc.** The **AD500** and **AD100** are 12-bit analog input boards, and the **DG24** is a digital I/O board. All three boards are designed to fit into the PC's short slot. The AD500 has eight channels of 12-bit resolution and an on-board amplifier with software programmable gains of 1, 10, and 100. It can digitize analog input signals at 7.5 conversions per second—providing very high immunity to 60-cycle line noise. The software-selectable analog input ranges are ±5 volts, ±500 millivolts, and ±50 millivolts. The inputs are single-ended and are protected to 35 volts. Seven TTL-compatible digital lines can be configured as inputs or outputs and can be used to activate relays, sense switch closures, or control other external devices. The AD100 is a single-channel version of the AD500 with three additional I/O lines. The DG24 is a digital I/O board based on the 8255 peripheral interface chip with 24 TTL-compatible lines that can be configured in software as input or output lines. The board can be configured to support handshaking. Interrupts are jumper-selectable. AD500, $239; AD100, $149; DG24, $95.
Real Time Devices, Inc., 1930 Park Forest Avenue, P.O. Box 906, State College, PA 16804; 814/234-8087

CIRCLE 317 ON READER SERVICE CARD

A 16-MHz, 80386-based microcomputer has been introduced by **Multitech Electronics, Inc.** The **Multitech 1100** base system comes standard with 1MB of RAM on the motherboard, a 1.2MB diskette drive, a 40MB hard-disk drive with a 28-millisecond average access time, a diskette/hard-disk controller board, a battery-backed clock/calendar, one parallel and two serial ports, and a 101-key keyboard. The Multitech 1100 has space for five half-height 5¼-inch storage devices, a socket for an 80287/80387 numeric coprocessor, and eight expansion slots (two PC/XT-type, five PC/AT-type, and one 32-bit dedicated memory bus). Multitech 1100, $3,999; with 80MB hard disk, $5,995; with 130MB hard disk, $6,495; Multitech enhanced color graphics board, $379; enhanced color display, $599.
Multitech Electronics, Inc., 1012 Stewart Drive, Sunnyvale, CA 94086; 408/773-8400

CIRCLE 302 ON READER SERVICE CARD

The **GV-386**, a 16-MHz, 80386-based machine, has been introduced by **PC Designs, Inc.** Operating with zero wait states when the cache is enabled, the GV-386 uses standard DRAM rather than the static column RAM. The GV-386 meets the I/O bus timing of the 8-MHz PC/AT, to provide maximum compatibility with existing add-on boards. Some features include 64KB, 35-nanosecond (ns) to 45-ns static RAM cache circuit, an on-board cache circuit that can be enabled/disabled via keyboard toggle, switchable 8-MHz/16-MHz clock speed, and a specially designed AT-compatible BIOS that allows for operation of BIOS timing loops independent of CPU clock speeds. Bundled with the GV-386 are Quarterdeck Office Systems' expanded memory manager and DESQview version 1.3 multitasking environment. GV-386 with 40MB hard disk, $3,950; with 80MB hard disk, $4,325.
PC Designs, Inc., 2500 N. Hemlock Circle, Broken Arrow, OK 74012; 918/251-5550

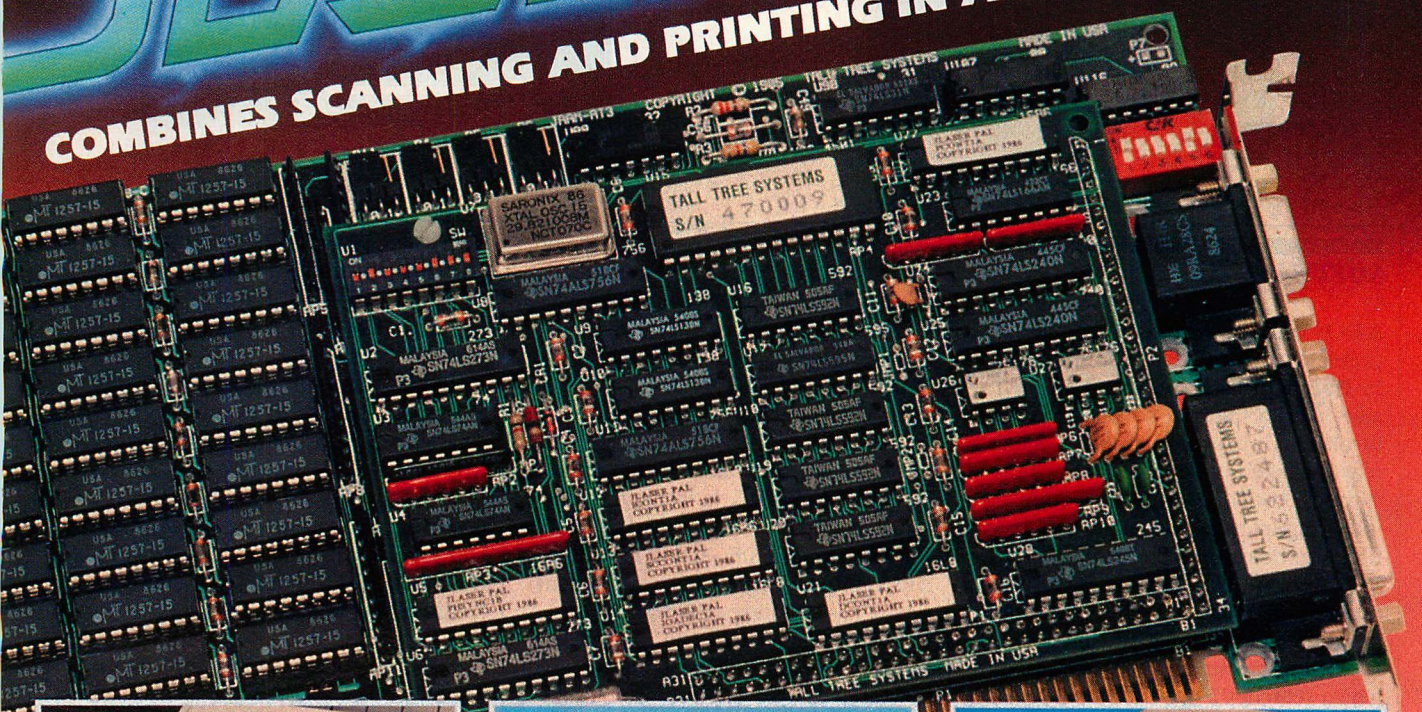
CIRCLE 305 ON READER SERVICE CARD

A coprocessor board for the PC bus that contains four 5-MIPS (million instructions per second) NEC μPD7281 data-flow (token-passing) processors is available from **Data Flow Imaging, Inc.** In addition to the four data-flow processors, the **DF-1** board contains an NEC μPD9305 memory access and general bus interface chip and 128KB of local image memory. The μPD7281 processor chip (nicknamed ImPP, for image-pipelined processor) uses a 17-bit-by-17-bit hardware multiplier for high processing speed. A single ImPP can compute a 3-by-3 convolution of a 512-by-512-pixel (8 bits per pixel) image in 2.98 seconds; two ImPPs will do the same convolution in 1.5 seconds. DF-1 supports direct memory access (DMA) transport of data between the PC memory and the DF-1 image memory, and can be configured to use interrupt-driven communications with the PC's CPU. DF-1, $995.
Data Flow Imaging, Inc., P.O. Box 116, Westwood, NJ 07675; 201/666-7970

CIRCLE 309 ON READER SERVICE CARD

JLASER PLUS

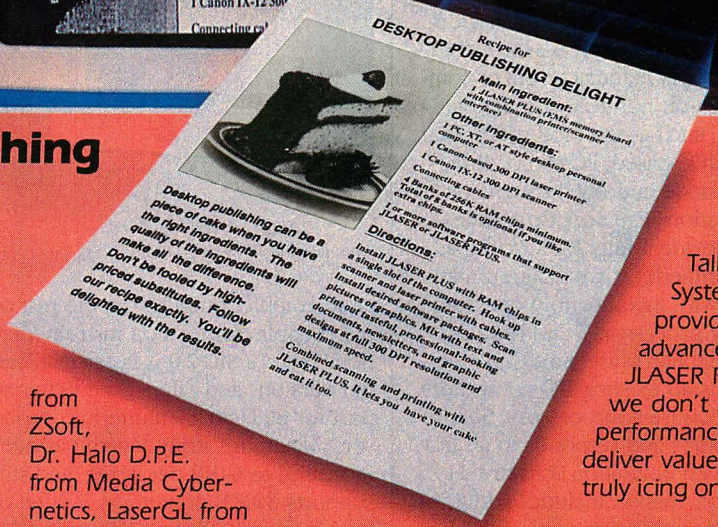
COMBINES SCANNING AND PRINTING IN A SINGLE BOARD!











It makes desktop publishing a piece of cake!

Tall Tree Systems introduces another breakthrough in desktop publishing with JLASER PLUS. We've combined a 2 MB EMS memory board and an interface to both a Canon®-based laser printer and scanner. JLASER PLUS increases the performance of both devices and gives you a low-cost solution to the limitations you've been experiencing with them.

Furthermore, the same memory that is made available to your printer and scanner is also available for all your other conventional applications. You get system memory, expanded LIM memory, extended memory in an AT-type machine, RAM Disk and print spooler — all in a single slot!

Supporting JLASER PLUS is a host of software packages, such as PC Paintbrush + from ZSoft, Dr. Halo D.P.E. from Media Cybernetics, LaserGL from Software Express, Ventura Publisher from Xerox, Page Builder from White Sciences, Le Print from Le Baugh Software, Fancy Font and Fancy Word from SoftCraft, Inc., and many more to be announced.

It takes a technological innovator like Tall Tree Systems to provide a major advancement like JLASER PLUS. And we don't stop at performance. We also deliver value, which is truly icing on the cake.

TALL TREE SYSTEMS
1120 San Antonio Road
Palo Alto, CA 94303
(415) 964-1980

CIRCLE NO. 194 ON READER SERVICE CARD

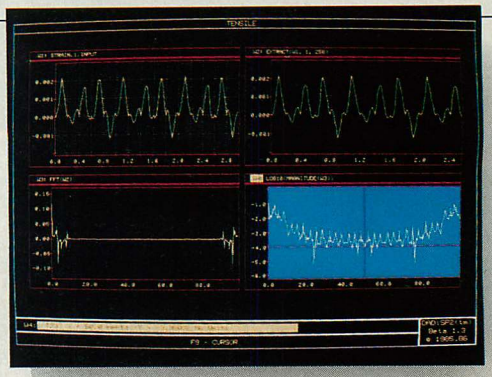
TALL TREE SYSTEMS

Canon is a registered trademark of Canon, Inc. All software packages listed are trademarks of their respective companies.

©1986 by Tall Tree Systems. All rights reserved.



A DOS extender and Command Interpreter from Polytron Corporation



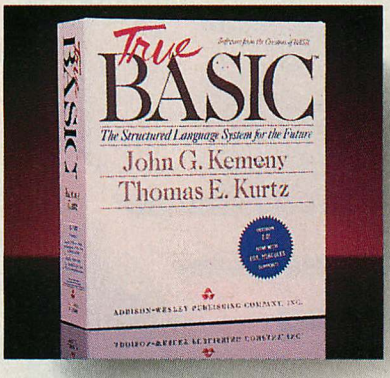
DADiSP Worksheet screen from DSP Systems

SOFTWARE

A DOS extender and command interpreter, **PolyShell**, has been introduced by **Polytron Corporation**. PolyShell adds a UNIX interface to DOS and is invoked as a program under DOS that may be exited or restarted at any time without rebooting. Any DOS command, including internal commands and batch files, can be called from within PolyShell, as can UNIX-like commands. This program includes more than 50 utility programs such as **split**, which splits large files into smaller files; **whereis**, which searches a drive for a particular file; **diff**, which compares two files and prints their differences; and **grep**, which searches files for character strings. $149.
Polytron Corporation, 1815 N.W. 169th Place, Suite 2110, Beaverton, OR 97006; 800/547-4000; in Oregon, 503/645-1150
CIRCLE 334 ON READER SERVICE CARD

Phoenix Technologies, Ltd. has developed a software coprocessor technology that enables manufacturers of computers based on non-IBM-compatible processors, such as the Motorola 68000, to offer a DOS-based, PC-compatible operating mode entirely through software emulation. The **Soft Co-Processor** will provide a fully PC/XT-compatible environment for workstation manufacturers with incompatible operating systems (or processors), and requires no additional hardware. Soft Co-Processor supports all existing I/O controllers. As with hardware coprocessors, Soft Co-Processor traps and responds to all I/O functions with emulation software. However, the soft version includes additional emulation software that remaps the Intel processor instruction set to the host system's native processor. PC video is supported through the standard screen or window manager. The DOS file system is fully integrated with the native file system, thus allowing users to read and write files created from either environment. Licensing to OEMs is available.
Phoenix Technologies, Ltd., 320 Norwood Park South, Norwood, MA 02062; 617/769-7020
CIRCLE 328 ON READER SERVICE CARD

True BASIC, Inc. has released **version 2.0** of **True BASIC**, in which the full range of graphics display cards, including Hercules, IBM EGA, and IBM CGA, is supported; GKI (graphic kernel system) syntax has been expanded; and mouse support added. True BASIC 2.0 features separately compilable modules that may be stored as libraries. Because modules have their own initialization sections, the user can set up global variables, separate graphics output from the rest of the program, or turn on instrumentation. True BASIC 2.0 is faster than earlier versions and offers even better support for 8087/80287 numeric coprocessors. Maximum string length has been increased to 64KB per string, and still supports 640KB of memory. Added debugging tools aid in tracing program execution or printing a cross-referenced listing. $149.90; upgrade, $30.00.
True BASIC, Inc., 39 S. Main Street, Hanover, NH 03755; 603/643-3882
CIRCLE 332 ON READER SERVICE CARD

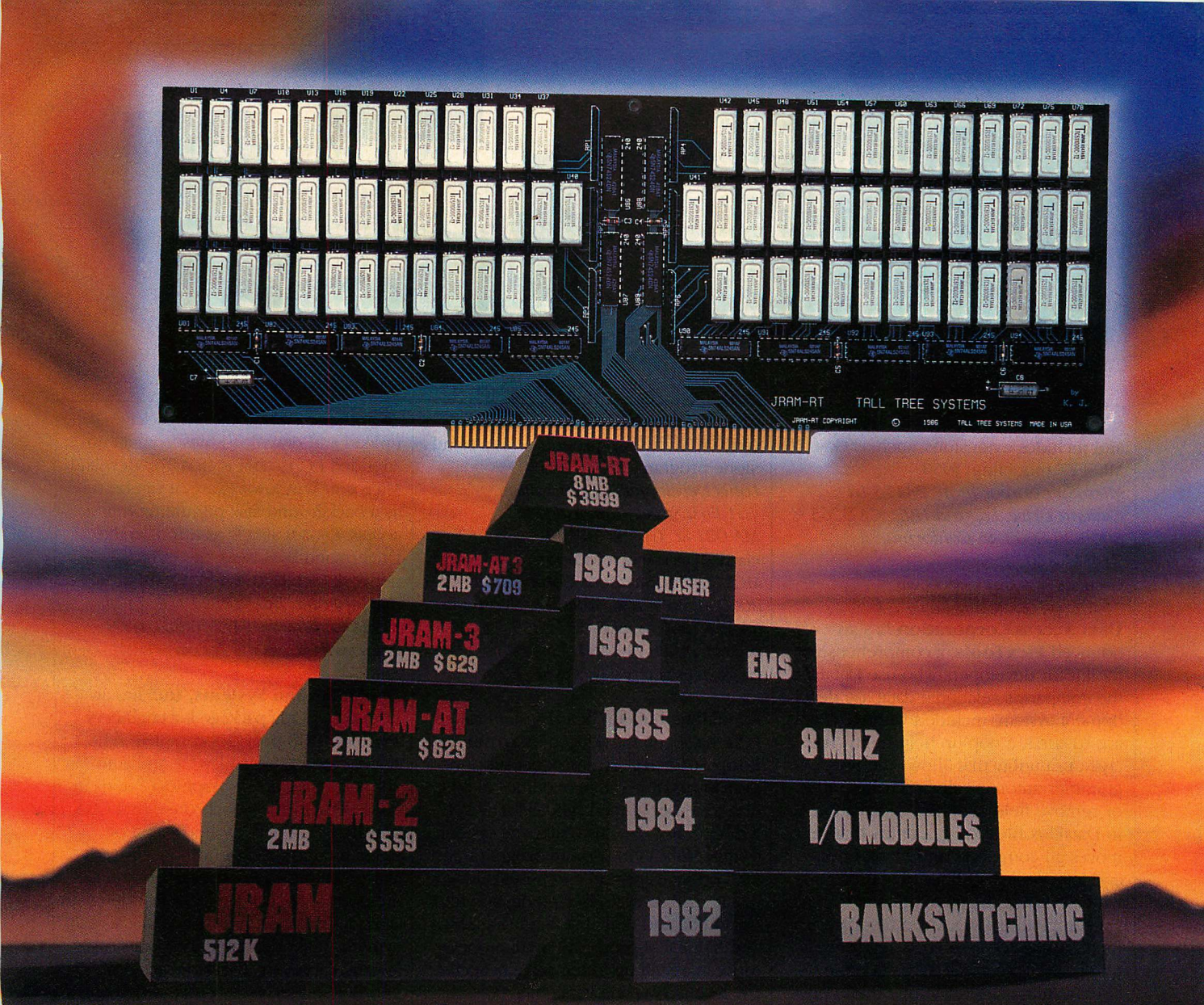


Version 2.0 from True BASIC, Inc.

Version 1.03 of the **DADiSP Worksheet**, a technical spreadsheet for digital signal analysis, is available from **DSP Systems**. DADiSP 1.03 features the **DSP PIPELINE**, which allows users to run external programs within the DADiSP environment. Using DSP PIPELINE, the user can, with a single command, export data from any DADiSP window to an external set of analysis or filtering algorithms and bring the modified data back into DADiSP. He also can run third-party IEEE 488, RS-232, and plotter drivers from DADiSP. $795.
DSP Systems, One Kendall Square, Cambridge, MA 02139; 617/577-1133
CIRCLE 324 ON READER SERVICE CARD

A serial communications product that runs concurrently with DOS and other applications on the PC family is being offered by **Sundance Software, Inc.** **HANDSHAKE** allows users to communicate over serial lines without leaving an application or waiting for the communications to finish before continuing. HANDSHAKE also allows several PCs to be connected to share resources without the need for a file server. Two PC HANDSHAKE links, $229; HANDSHAKE for each additional PC, $99.
Sundance Software, Inc., P.O. Box 434, Redmond, WA 98073; 206/885-0759
CIRCLE 327 ON READER SERVICE CARD

MicroSim Corporation has released **version 3.0** of its **PSpice** analog electrical-circuit simulator. PSpice has been converted from FORTRAN into C in order to make use of the advantages of C: A reduction in the program code size (and more compact data structures) has doubled the maximum circuit size from 120 to 240 metal oxide semiconductors (MOS) transistors on the PC. Execution times have been improved. Changes in input processing allow circuit nodes to be given names or numbers, thus facilitating access nodes inside subcircuits. PSpice 3.0 is compatible with (Univer-



TALL TREE SYSTEMS.
A Technological Innovator.
Always a Step Ahead!

For true industry leadership, look no further than Tall Tree Systems.

We have a history of being first.

We were the first to introduce bankswitching. The first with two megabyte memory boards. The first with I/O modularity in a single slot. The first with 8 MHz speed capabilities. The only maker of single command EMS boards. The first with a laser printer solution — JLASER — that allows you to do full-page graphics and multiple type fonts on any Canon® or Ricoh® laser engine.

Now, we're first again with memory expansion for the IBM®RT.

Innovation is our tradition.

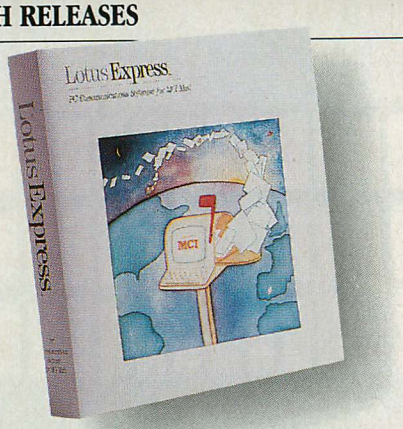
Our trademark is superior technology at the lowest possible price.

TALL TREE SYSTEMS

1120 San Antonio Road • Palo Alto, CA 94303 • (415) 964-1980

CIRCLE NO. 197 ON READER SERVICE CARD

©1986 by Tall Tree Systems. All rights reserved. IBM, RT are registered trademarks of International Business Machines Corp. Canon and Ricoh are registered trademarks of Canon Corp. and Ricoh Corp., respectively.



Lotus Express for MCI Mail from Lotus and MCI



PCTERM, a multiterminal emulator from Crystal Point, Inc.

sity of California) Berkeley SPICE2G.6 and includes extensions, such as GaAs MESFET modeling. $950.
MicroSim Corporation, 23175 La Cadena Drive, Laguna Hills, CA 92653; 800/826-8603, in California, 714/770-3022

CIRCLE 326 ON READER SERVICE CARD

Data Access Corporation has announced **FlexTools**, a comprehensive toolkit for developers who use DataFlex, the company's relational database development system. Included with FlexTools are FLEXHELP, a pop-up help utility; FLEXCHANGE, a utility that allows the programmer automatically to change, add, or remove field or index information in DataFlex files; STRIP, a utility that removes all comments and blank lines and left-justifies all lines of source code, which renders the code difficult to read, yet compilable; RELOCATE, a file-management utility; FIXREF, a cross-reference utility; and FTREF2, a source code file search utility. The revamped menu environment lets the user define as many as 24 selections per menu; in addition, the system detects when a menu has more than 12 selections and automatically splits the screen. $195.
Data Access Corporation, 8525 S.W. 129th Terrace, Miami, FL 33156; 305/238-0012

CIRCLE 329 ON READER SERVICE CARD

Lotus Development Corporation and **MCI Communications Corporation** have announced their plans to jointly market **Lotus Express for MCI Mail**, a communications product that makes possible quick and reliable exchange of any binary file. The integrated program gives the user a Lotus interface to the MCI Mail network and its full range of electronic and off-network delivery options, along with on-line help and mail-management facilities. The program lets users compose, read, send, file, archive, and print electronic mail directly from their PCs. Lotus Express can run in the background or as a standalone program. $100.
Lotus Development Corporation, 161 First Street, Cambridge, MA 02142; 617/577-1100

CIRCLE 341 ON READER SERVICE CARD
MCI Communications Corporation, 2000 M Street NW, Suite 300, Washington, DC 20036; 800/624-2255; in Washington, 202/833-8484

CIRCLE 321 ON READER SERVICE CARD

A multiline bulletin board system called **DXL** has been introduced by **Inner Loop Software**. This product supports as many as nine telephone lines in addition to a user at the keyboard. The system operator can customize the system to the application without programming. DXL provides private and public electronic mail, user-to-user mode, and file uploading and downloading using ASCII and XMODEM protocols. System operators can assign up to ten account security levels. Two-line version, $200.
Inner Loop Software, 5456 McConnell Avenue, Suite 120, Los Angeles, CA 90066; 213/822-2800

CIRCLE 322 ON READER SERVICE CARD

***PC*TERM 3.1**, a multiterminal emulator, has been announced by **Crystal Point, Inc.** *PC*TERM is designed to communicate with a wide spectrum of computers and information services; it currently emulates 15 asynchronous terminals including the IBM 3101. *PC*TERM can be custom programmed to create unique menu environments; its programming language has more than 80 communications-related commands that allow this product to emulate not only a terminal, but an operator as well, for repetitive data entry. It can be programmed for unattended data transfers with full error-recovery capability and conditional testing based on response. *PC*TERM also permits unattended remote access to files on the PC, and remote users of *PC*TERM and other communications packages can upload and download files from an unattended PC. *PC*TERM is IBM TopView and Microsoft Windows aware; it can be run as a background task permitting remote access. *PC*TERM also is featured on the Ungermann-Bass Net/One Async PC network (see Hardware). $250.
Crystal Point, Inc., 12221 N.E. 140th Kirkland, WA 98034; 206/821-1909

CIRCLE 336 ON READER SERVICE CARD

Dasoft Design Systems, Inc. has announced a CAD package for engineers. **Project: PCB** provides complete schematic capture, board layout, and autorouting capabilities. Graphics options include Hercules monochrome, IBM CGA, and IBM EGA with monochrome or color monitor. A system mouse is required, and a wide variety of plotters is supported. Special features include user-definable schematic and silk-screen symbols; user-definable footprints to 1/100-inch accuracy; block move, copy, and delete functions; screen display of unrouted traces on the board; a tag-and-drag function in interactive routing mode; four-layer boards (two signal layers, plus internal planes); single-net or full-board autorouting; routines for producing net lists, parts lists, and hole lists; plot routines for camera-ready artwork; and optional predefined libraries of parts and symbols. $950.
Dasoft Design Systems, Inc., P.O. Box 8088, Berkeley, CA 94707-8088; 415/486-0856

CIRCLE 323 ON READER SERVICE CARD

Command Plus version 1.1 has been announced by **ESP Software Systems, Inc.** Created as a programming environment for DOS, Command Plus features an aliasing command, which allows the user to create fast, memory-resident command macros; a history command, which lets the user edit and execute commands that were run as many as 48 command lines before; disk-manage-

A Challenge to Microsoft® C...

We challenge Microsoft C (Ver 4.0) to a C compiler duel to the finish, measuring compile, link, and execution times. If they win, we will stop advertising for two months.

by Roy Sherrill

If Microsoft C (Ver 4.0) can beat Optimum-C then we will stop advertising in all magazines for two full months and, win or lose, we will publish the results in its entirety. Even the Microsoft ads say "The Fastest C you've ever seen," so let the challenge begin.

Walter says Optimum-C is better

It all started when Walter Bright, the developer of Optimum-C, was explaining his new global optimizing C compiler and how it's code would be faster than Microsoft C (Ver. 4.0). Walter and I were frustrated because here we had a C compiler that would beat Microsoft C on 7 out of 10 benchmarks and also compile and link faster; yet our marketing consultant, Mark Astengo, told us that Microsoft C had a lock on the C compiler market and by 1990 they would probably have an 80% market share. Then Mark said, "Roy, if your C compiler is as fast as you say it is, why not challenge Microsoft C to a duel? If Microsoft wins, Datalight should stop advertising for two months and print the results of the test, win or lose." Well, I've always been one for a challenge. So here it is...

We only ask the following...

The benchmark suite will consist of the set of programs that Microsoft supplied to *Computer Language* for their February 1987 C compiler review issue. Microsoft will make available the programs to Datalight at least two weeks prior to the benchmarking. The benchmarking will be between Microsoft C 4.0 and Optimum-C. It will occur at a mutually agreed upon time and place. Interested individuals will be allowed to attend. The benchmarks will be compiled and run on a standard IBM PC-AT.

There will be two separate tests for each program: compile and link speed, and execution speed. For each test, a representative from each company will set up the compiler so that it performs at its best.

The benchmarks will be adjusted so that they take sufficiently long to run, that the tolerance involved in timing them is insignificant. The winner is determined by the compiler with the faster execution times for the majority ofthe benchmarks. We'd like an answer from Microsoft no later than April 1, 1987.

So what's a global optimizer?

A global optimizer looks at an entire function at once, analyzing and optimizing the whole function. A technique called data flow analysis is used by Optimum-C to gather information about each function. This enables your compute-bound programs to execute as much as 30% faster after global optimization. But, there is one catch...because the global optimizer ruthlessly searches for ways to speed-up execution speed and minimize memory usage, it has relatively slow compile times. No need to worry, though, because you can merely turn the global optimizer off. In fact, you can select all, none, or partial of the following optimizations: constant propagation, copy propagation, dead assignment elimination, dead variable elimination, dead code elimination, do register optimizations, global common subexpression elimination, loop invariant removal, loop induction variables, optimize for space, optimize for time, very busy expressions.

Choose from five memory models

Speed your programs by selecting the memory model that best suits your application.

Memory Models

Model	Code	Data
Compact	64k total code & data	
Small	64k	64k
Program	1M	64k
Data	64k	1M
Large	1M	1M

Compiling, one step...

Now with the one step DLC program you can create .OBJ, .EXE and .COM files. Also, DLC can handle multiple files and run MASM on your assembly files.

Try Optimum-C risk free

Try Optimum-C for 30 days and if you are not 100% satisfied return it for a full refund. Also, for a limited time we are including a *free C tutorial which is a combination workbook and floppy disk to help lead you through the C language with tutorials, quizzes, and program exercises.

O.K. Microsoft, it's up to you. We've put two months of advertising on the line that says you can't beat Optimum-C to a real test. Your answer, please?

PRICES

Developer's Kit	still only $99
Optimum-C	$139
(includes library source code)	

Add $5 for shipping in US/$15 outside US
COD (add $2.50)

Not Copy Protected




ORDER TOLL-FREE TODAY!

1-800-221-6630

Microsoft and MS-DOS are registered trademarks of the Microsoft Corporation.

Magazine Reviewers Shocked by DATALIGHT's Performance...

"Reviewing this compiler was quite a surprise for us. For such a low price, we were expecting a "lightweight" compiler. What we got was a package that is as good as or better than most of the "heavyweights." Datalight C implements a complete C language. It also compiles quickly, doesn't take up much disk space, and looks impressive in the benchmarks."

***DR. DOBBS*, August 1986**

"This is a sharp compiler!... what is impressive is that Datalight not only stole the compile time show completely, but had the fastest Fibonacci executable time and had excellent object file sizes to boot!"

***COMPUTER LANGUAGE*, February 1986**

Optimum-C Version 3.0

- Full UNIX System 5 C language plus ANSI extensions
- Fast/tight code via powerful optimizations including common sub-expression elimination
- DLC one-step compile/link program
- Multiple memory model support
- UNIX compatible library with PC functions
- Compatible with DOS linker and assembler
- Third-party library support
- Automatic generation of .COM files
- Supports DOS pathnames, wild cards, and Input/Output redirection
- Compatible with Lattice C version 2.x
- Interrupt handling in C
- Debugger support
- ROMable code support/start-up source

MS-DOS® Support Features

- Mouse support
- Sound support
- Fast screen I/O
- Interrupt handler

MAKE Maintenance Utility

- Macro definition support
- MS-DOS internal commands
- Inference rule support
- TOUCH date manager

Tools in Source Code

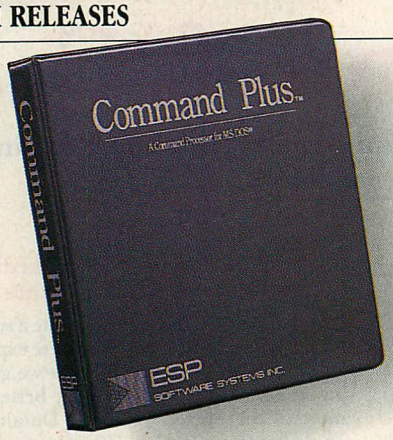
- cat—UNIX style "type"
- diff—Text file differences
- fgrep—fast text search
- pr—Page printer
- pwd—Print working directory
- wc—Word count

Datalight

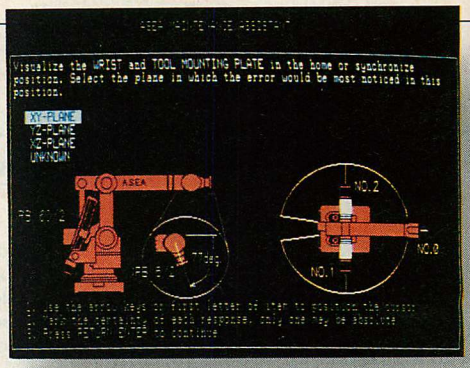
Box 82441
Kenmore, Washington 98028
(206) 367-1803

*Limited offer available exclusively to readers who purchase directly from Datalight.

CIRCLE NO. 107 ON READER SERVICE CARD



Command Plus programming environment from ESP Software Systems



Screen from Personal Consultant Plus (PC Plus) from Texas Instruments

ment utilities, which contain options to process subdirectories recursively, sort files, and specify multiple source files; BROWSE, a full-screen file viewer with regular expression searching and IBM EGA 43-line support; command line editing with user-definable editing keys; and the ability to create multiple commands and arguments on a line. $79.95.
ESP Software Systems, Inc., 11965 Venice Blvd., Suite 309, Los Angeles, CA 90066; 800/992-4377; in California, 213/390-7408

CIRCLE 325 ON READER SERVICE CARD

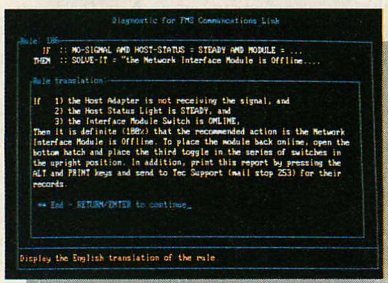
A BASIC programmer's package has been introduced by **MicroHelp, Inc.** **Sta-Res** makes compiled BASIC programs memory-resident, such that they can be invoked by pressing a selectable hot key. For machines that use DOS 3.0 or later plus an expanded/extended memory board, adding the EMS/RAM disk module ensures that each memory-resident program takes only 7KB of DOS memory, and the balance of the program is swapped in and out of memory as needed. Included in the program is a SHELL facility that allows the user to execute any program, even other BASIC programs and the BASIC interpreter. Sta-Res is not copy protected, and no royalties are imposed when Sta-Res-assisted programs are distributed as a part of executable programs. $95; EMS/RAM disk module, $50.
MicroHelp, Inc., 2220 Carlyle Drive, Marietta, GA 30062; 404/973-9272

CIRCLE 339 ON READER SERVICE CARD

Big Bang Software, Inc. has released an **MC68000/68010 Simulator/Debugger** for the PC. The package enables the user to test and debug 68000/10 software on the PC; hexadecimal files of Motorola S-Records can be disassembled and displayed. Instructions can be executed in fast or single-step mode, and the results of each executed instruction on registers, flags, and 68000/10 memory are immediately available for display. During single-stepping, the display shows the last instruction executed, the current contents of all registers, and the instruction following the last executed. All 68000/10 instructions, addressing modes, and condition codes are fully supported. Load, dump, and breakpoint facilities are included. $285.
Big Bang Software, Inc., P.O. Box 879, Panama City, FL 32402; 904/784-3393

CIRCLE 333 ON READER SERVICE CARD

The **Personal Consultant Series**, a family of expert system development tools is being offered by **Texas Instruments** (TI). **Personal Consultant Easy** is designed for users just beginning in expert system development, and it features high functionality for rapid prototyping of small- to medium-sized expert system applications. **Personal Consultant Plus** (PC Plus), a more powerful member, is designed to take



Screen from TI's Personal Consultant Easy

advantage of PC/AT-class machines. PC Plus provides extended knowledge-representation features, increased rule capacity, and access to LISP, thus providing sophisticated developers the flexibility to customize their applications. Knowledge bases created with Personal Consultant Easy are upwardly compatible with PC Plus. Applications developed under either product can be delivered on microcomputers with the addition of an optional runtime diskette. Personal Consultant Easy, $495; PC Plus, $2,950; demo diskette, $25; runtime diskettes, $95 each or $995 for 20, PC Scheme (TI's LISP compiler), $95.
Texas Instruments, Data Systems Group, AI Software Marketing Department, M/S 2244, P.O. Box 2909, Austin, TX 78769-2909; 800/527-3500

CIRCLE 331 ON READER SERVICE CARD

Unicorn Systems Company has announced that **MicroCICS** for the PC-370 now supports CICS release 1.7. MicroCICS, provides a full CICS development environment, allows programmers to enter, compile, test, debug, and execute CICS applications without accessing the mainframe until the completed code is ready for uploading. MicroCICS support for CICS 1.7 includes modifications to the Command Language Translator, Command Routines, and debugging tools that provide integrated support for release 1.7 features. MicroCICS also can be used to write programs compatible with CICS 1.5 or 1.6.

MicroCICS will interpret and execute applications according to the features supported by the CICS release that the programmer selects. It allows development of command-level CICS COBOL and assembly language source programs outside the mainframe environment. MicroCICS includes on-line screen generation and automatic creation of basic mapping support (BMS) maps, creation and maintenance of test data files, program execution and testing using a powerful symbolic debugger, and host communications for program uploading and downloading. PC-370, $4,495.
Unicorn Systems Company, 3807 Wilshire Blvd., Los Angeles, CA 90010; 213/380-6974

CIRCLE 340 ON READER SERVICE CARD

The material that appears in Tech Releases is based on vendor-supplied information. These products have not been reviewed by the PC Tech Journal *editorial staff.*

Unbiased Advice.

Our friendly, non-commissioned salespeople are always prepared to assist you. We also have experienced technical consultants who can answer questions, help you compare products and send you detailed product information tailored to your needs. Since we're not affiliated with any software publisher or manufacturer, we'll give you an unbiased look at the products we carry.

High Quality.

We stock hundreds of high quality software development tools specifically for IBM personal computers and compatibles. And as new products become available, we'll sell only those that meet our high standards for quality and value.

Manufacturer Support.

The products we sell are the latest versions and come with the same technical support as if buying directly from the manufacturer.

Return Guarantees.

Our goal is customer satisfaction and that's why we offer a 30-day documentation evaluation period or a 30-day return guarantee on most of our products. Please call for specific details.

Immediate Shipment.

Most products are in stock and are ready for shipment from our large inventory.

Discounts.

You'll save money on all of your software purchases from Programmer's Connection. Our ads show both the discount and retail prices for each product so you'll always know exactly how much you'll save.

FREE Shipping.

Shipping is FREE if you have your order shipped via standard UPS anywhere in the USA. We can also express your order to you with no special fees and we'll only charge you the shipping carrier's standard rate. Many other companies profit from overcharges plus special fees for express shipments.

Credit Cards.

We'll charge your credit card at the time we ship your order. Other companies may charge your credit card at the time they take your order so they can use your money interest-free while you wait for your shipment.

No Sales Tax.

Customers outside of Ohio are not charged state sales tax. Ohio residents are charged 6 percent.

No Hidden Charges.

Quite simply, the prices you see on the next two pages are all you pay. We don't charge extra for standard UPS shipping, credit cards, COD orders, purchase orders or special handling.

Turn the page for our latest advertised price list.

PROGRAMMER'S CONNECTION

When you need programmer's development tools, Programmer's Connection is your best one-stop source. We've specialized in development software for IBM personal computers since 1984 and are experienced in providing a full range of quality products and customer services.

At Programmer's Connection, you get all of the benefits of buying directly from the manufacturer and none of the drawbacks. So call us today and discover the advantages of our one-stop service for yourself. You'll be glad you did!

programmer's connection

apl language

APL*PLUS/PC *by STSC*	595	429
APL*PLUS/PC Spreadsheet Mgr *by STSC*	195	139
APL*PLUS/PC Tools Vol 1 *by STSC*	295	199
APL*PLUS/PC Tools Vol 2 *by STSC*	85	59
Financial/Statistical Library *by STSC*	275	195
Pocket APL *by STSC*	95	69
STATGRAPHICS *by STSC*	795	579

artificial intelligence

1st-CLASS *by Programs in Motion*	495	399
APT *from Solution Systems*	65	CALL
Arity Combination Package	1225	1119
Expert System Development Pkg	295	259
File Interchange Toolkit	50	45
PROLOG Compiler & Interpreter	795	699
Screen Design Toolkit	50	45
SQL Development Package	295	259
Arity PROLOG Interpreter	350	309
Arity Standard Prolog	95	79
AutoIntelligence *by IntelligenceWare*	990	CALL
ExpertEDGE Advanced *by Human Edge*	2500	CALL
ExpertEDGE Professional *by Human Edge*	5000	CALL
Experteach II *by IntelligenceWare*	475	359
EXSYS *Development Software by EXSYS*	395	319
EXSYS Runtime System *New*	600	479
GCLISP *Golden Common LISP by Gold Hill*	495	CALL
GCLISP 286 Developer *by Gold Hill*	1190	CALL
Insight 1 *by Level Five Research*	95	75
Insight 2+ *by Level Five Research*	485	379
Intelligence/Compiler *IntelligenceWare*	990	749
Logic-Line Series 1 *by Thunderstone*	90	85
Logic-Line Series 2 *by Thunderstone*	125	115
Logic-Line Series 3 *by Thunderstone*	150	139
LPA microPROLOG *All Varieties*	CALL	CALL
Microsoft LISP *Common LISP*	250	163
MPROLOG Language Primer *LOGICWARE New*	50	45
MPROLOG P500 *by LOGICWARE New*	495	395
MPROLOG P550 *by LOGICWARE New*	220	175
PC Scheme *by Texas Instruments*	95	84
Personal Consultant Easy *by TI*	495	435
Personal Consultant Plus *by TI*	2950	2589
Personal Consultant Runtime	95	85
QNIAL *by NIAL Systems*	375	349
TransLISP *from Solution Systems*	95	CALL
TransLISP PLUS *from Solution Systems New*	195	CALL
Turbo PROLOG *by Borland Intl New Version*	100	65

assembly language

386 ASM/LINK *Cross Asm by Phar Lap*	495	389
8088 Assembler *w/Z-80 Trans by 2500 AD*	100	89
ASMLIB *Function Library by BC Assoc*	149	129
asmTREE *B-Tree Dev System by BC Assoc*	395	339
Cross Assemblers *Various by 2500 AD*	CALL	CALL
Microsoft Macro Assembler	150	95
Norton Utilities *by Peter Norton*	100	59
Turbo EDITASM *by Speedware*	99	84
Uniware Cross Assemblers *Various by SDS*	295	249
Visible Computer: 8088 *Software Masters*	80	65

basic language

BetterBASIC *by Summit Software Clearance*	200	119
EXIM Services Toolkit *by EXIM*	CALL	CALL
Finally *by Komputerwerks*	99	85
Inside Track *from Micro Help*	65	51
MACH 2 *by Micro Help*	75	59
Microsoft QuickBASIC	99	65
87 QB Pak *by Hauppauge New*	69	59
Peeks 'n Pokes *from MicroHelp*	45	37
Professional BASIC *by Morgan*	99	75
8087 Math Support	50	42
Stay-Res *by MicroHelp*	95	74
True Basic *New Version*	150	99
True Basic *w/Run-time*	245	179
BASICA Converter	50	45
Run-time Module	150	99
Various Other Utilities	50	45
Turbo BASIC *by Borland Intl New*	100	69

blaise products

ASYNCH MANAGER *Specify C or Pascal*	175	124
C TOOLS PLUS	175	124
EXEC *Program Chainer*	95	75
LIGHT TOOLS *for Datalight C New*	100	89
PASCAL TOOLS	125	99
PASCAL TOOLS 2	100	79
PASCAL TOOLS & PASCAL TOOLS 2	175	124
RUNOFF *Text Formatter*	50	45
TURBO ASYNCH PLUS	100	79
TURBO POWER TOOLS PLUS	100	79
VIEW MANAGER *Specify C or Pascal*	275	189

borland products

EUREKA *Equation Solver New*	100	69
REFLEX & REFLEX Workshop	200	129
REFLEX *Data Base System*	150	89
REFLEX Workshop	70	45
Turbo BASIC *New*	100	69
Turbo DATABASE TOOLBOX	70	47
Turbo EDITOR TOOLBOX	70	47
Turbo GAMEWORKS TOOLBOX	70	47
Turbo GRAPHIX TOOLBOX	70	47
Turbo LIGHTNING	100	64
Turbo Numerical Methods Library *New*	100	69
Turbo PASCAL and TUTOR *New*	125	85
Turbo PASCAL *with 8087 and BCD*	100	64
Turbo TUTOR	40	28
Turbo Prolog Compiler *New Version*	100	64
Turbo Prolog Toolbox *New*	100	64
Word Wizard	70	47
Word Wizard and Turbo Lightning	150	94

c++

C++ *by Guidelines w/kernel 1.1*	195	179
PforCe++ *Library for C++ by Phoenix New*	395	229

c compilers

68000/10/20 Cross Compiler *by SDS*	595	CALL
C86PLUS *by Computer Innovations New*	497	CALL
Datalight C Compiler *Small Model*	60	49
Datalight Developer Kit	99	79
Datalight Optimum-C *New*	139	119
DeSmet C *w/Debugger*	159	138
DeSmet C *w/Debugger & Large Case*	209	184
Eco-C *Development System by Ecosoft*	125	83
Lattice C Compiler *from Lattice*	500	275
Mark Williams Let's C Combo Pack *New*	125	99
Let's C *Compiler*	75	57
csd *Source Level Debugger*	75	57
Mark Williams MWC-86	495	289
Microsoft C *with CodeView*	450	275
Wizard C Combo *by Wizard Systems*	750	529
Wizard C Compiler	450	299
ROM Development Pkg	350	259

c interpreters

C-terp *by Gimpel, Specify compiler*	300	235
C Trainer *with Book by Catalytix*	122	87
Instant C *by Rational Systems*	500	379
Introducing C *by Computer Innovations*	125	CALL
Run/C *from Lifeboat*	150	88
Run/C Professional *from Lifeboat*	250	159

c utilities

C Essentials *by Essential Software*	100	75
C-ISAM *by Informix*	225	195
C to dBase *by Computer Innovations*	150	CALL
c-tree & r-tree Combo *by FairCom*	650	529
c-tree *ISAM File Manager*	395	329
r-tree *Report Generator*	295	249
C Utility Library *by Essential*	185	135
C Windows *by Syscom*	100	85
C Wings *by Syscom*	50	43
CI ROMPac *by Computer Innovations*	195	CALL
dbQUERY *All Varieties by Raima*	CALL	CALL
dbVISTA *Single-User DBMS by Raima*	195	155
with Source Code	495	425
dbVISTA *Multi-User DBMS by Raima*	495	425
with Source Code	990	845
dBx *dBase/C Translator by Desktop AI*	350	314
with Library Source Code	550	493
Entelekon Combo Package	200	169
C Function Library	130	109
C Windows	130	109
Superfonts for C	50	43
Essential Comm Library *w/Debugger*	250	195
Breakout Debugger *Any language*	125	99
Essential Comm Library	185	135
Essential Graphics *by Essential Software*	250	195
Flash-up Windows *by Software Bottling*	90	79
GraphiC *Mono v2.2 by Sci Endeavors*	280	209
GraphiC *Color v3.0 by Sci Endeavors*	350	284
GRAFLIB *by The Librarian*	175	CALL
Greenleaf Comm Library *by Greenleaf*	185	127
Greenleaf Data Windows *by Greenleaf*	225	157
with Source Code	450	295
Greenleaf Functions *by Greenleaf*	185	127
HALO *by Media Cybernetics*	300	209
HALO Development Pkg *for Microsoft New*	595	395
The HAMMER *by OES Systems*	195	139
HELP/Control *by MDS*	125	109
MetaWINDOWS *No Royalties*	185	115
MetaFONTS	80	58
MetaWINDOWS/Plus *by Metagraphics*	235	189
MetaFONTS/Plus	235	189
PANEL *by Roundhill Computer Systems*	295	215
PC Lint *by Gimpel Software*	139	99
PLOTHI *by The Librarian*	175	CALL
PLOTHP *by The Librarian*	175	CALL
Scientific Subroutine Lib *by Peerless*	175	134
screenplay *by Flexus*	175	129
Vector87 *by Vectorplex Data Systems*	150	135
Vitamin C *by Creative Programming*	225	CALL
VC Screen *Forms Designer*	100	82
Zview *by Data Management Consultants*	245	189

cobol language

Micro Focus COBOL Workbench	4000	CALL
Micro Focus Level II COBOL	1500	CALL
COGRAPHICS	250	CALL
COMATH	200	CALL
FORMS-2	300	CALL
Level II Animator	900	CALL
Level II SOURCEWRITER	2000	CALL
Micro Focus Level II COBOL *for Novell*	2000	CALL
Micro Focus Professional COBOL	3000	CALL
Multi-user Runtime *for PC Network*	500	CALL
Microsoft COBOL *See Microsoft Section*	CALL	CALL
Realia COBOL	995	785
Realia CICS *New*	995	785
RM/COBOL *by Ryan-McFarland*	950	639
RM/COBOL 85 *by Ryan-McFarland*	1250	895
screenplay *by Flexus*	175	129

debuggers & profilers

386 DEBUG *Cross Debugger by Phar Lap*	195	129
Advanced Trace-86 *by Morgan Computing*	175	125
CI Probe *by Computer Innovations*	225	CALL
Codesifter *Profiler by David Smith*	119	94
Codesmith-86 *by Visual Age*	145	99
DSD86 *by Soft Advances*	70	61
DSD87 *by Soft Advances*	100	79
MiniProbe *by Atron New*	395	CALL
Periscope I *by The Periscope Company*	345	CALL
Periscope II *w/NMI Breakout Switch*	175	CALL
Periscope II-X *Software only*	145	CALL
The PROFILER *with Source Code by DWB*	125	89
The WATCHER *Profiler by Stony Brook*	60	51

forth language

CFORTH *Native Code Compiler by LMI*	300	229
Forth/83 Metacompiler *Specify Target*	750	599
PC/Forth *by Laboratory Microsystems*	150	109
PC/Forth+ *by Laboratory Microsystems*	250	199
Advanced Color Graphics Support	100	74
Enhanced Graphics Support	200	148
Intel 8087 Support	100	74
Interactive Symbolic Debugger	100	74
Native Code Optimizer	200	148
Software Floating Point	100	74
UR/Forth *and support utilities by LMI*	CALL	CALL

fortran language

50 MORE: FORTRAN *by Peerless Engr*	125	95
ACS Time Series *Alpha Computer Service*	495	399
Btrieve *ISAM File Mgr by SoftCraft*	245	194
Essential Graphics *by Essential Software*	250	195
For-Winds *Alpha Computer Service*	90	69
Forlib-Plus *Alpha Computer Service*	70	49
FORTLIB *by Sutrasoft*	95	CALL
FORTRAN Addenda *by Impulse Engr*	95	85
FORTRAN Addendum *by Impulse Engr*	165	139
GRAFLIB *by Sutrasoft*	175	CALL
HALO *by Media Cybernetics*	300	209
I/O PRO *by MEF Environmental*	149	129
Microcompatibles Combo Package	240	219
Grafmatic	135	119
Plotmatic	135	119
Microsoft FORTRAN *w/CodeView - New Version*	450	CALL
No Limit *by MEF Environmental*	129	115
PANEL *Screen Designer by Roundhill*	295	215
PLOTHI *by Sutrasoft*	175	CALL
PLOTHP *by Sutrasoft*	175	CALL
RM/FORTRAN *Ryan-McFarland*	595	CALL
Scientific Subroutine Lib *by Peerless*	175	134
Statistician *Alpha Computer Service*	295	245
Strings & Things *Alpha Computer Service*	70	51
Vector87 *by Vectorplex Data Systems*	150	135

lattice products

Lattice C Compiler *from Lattice*	500	275
with Library Source Code	900	495
C Cross Reference Generator	50	37
with Source Code	200	145
C-Food Smorgasbord *Function Library*	150	95
with Source Code	300	184
C-Sprite *Source Level Debugger*	175	129
Curses *Screen Manager*	125	89
with Source Code	250	178
dBC *dBase File Manager for C*	250	178
with Source Code	500	356
LMK *Make Facility*	195	139
RPG II Compiler *No Royalties*	750	626
RPG II Combo *with SEU & Sort/Merge*	1100	939
RPG II Screen Design Aid Utility *New*	350	309
SecretDisk *File Encryption Utility*	120	89
SideTalk *Resident Communications*	120	89
SSP/PC *Scientific Library New*	350	269
Text Management Utilities	120	89
TopView Toolbasket *Function Library*	250	178
with Source Code	500	356

logitech products

LOGIMOUSE C7 *Specify Connector Type*	99	83
with PLUS Pkg	119	98
with PLUS & PC Paintbrush	169	134
with PLUS & CAD Software	189	153
with PLUS & Reflex	199	162
with PLUS & CAD & Paint	219	179
with PLUS & CAD & Paint & Reflex	299	245
LOGIMOUSE BUS *with PLUS Pkg New*	139	115
with PLUS & PC Paintbrush New	189	149
with PLUS & CAD Software New	209	175
with PLUS & Reflex New	219	179
with PLUS & CAD & Paint New	239	195
with PLUS & CAD & Paint & Reflex New	319	259
MODULA-2/86 Holiday Package	199	159
MODULA-2/86 Compiler	89	62
MODULA-2/86 with 8087 Support	129	98

MODULA-2/86 with PLUS Pkg	189	138
Library Sources	99	88
Make Utility	29	25
ROM Package	199	172
Run Time Debugger	69	56
Turbo to Modula Translator	49	42
Utilities Package	49	42
Window Package	49	42
REPERTOIRE *for MODULA-2/86 by PMI*	89	79
Object Code Only *New*	19	15

microport products

System V/AT *by Microport Systems*	440	395
Runtime System *(Operating System)*	159	145
Software Development System	169	155
Text Preparation System	169	155
User Upgrade *3 to Unlimited Users*	169	155

microsoft products

Microsoft BASIC Interpreter *for XENIX*	350	239
Microsoft C *with CodeView*	450	275
Microsoft COBOL Compiler	700	439
for XENIX	995	635
Microsoft COBOL Tools *with Debugger*	350	CALL
for XENIX	450	289
Microsoft FORTRAN *w/CodeView - New Version*	450	CALL
for XENIX	695	439
Microsoft Learning DOS	50	36
Microsoft LISP *Common LISP*	250	163
Microsoft MACH 10 *w/Mouse & Windows*	549	385
Microsoft MACH 10 *Board only*	399	285
Microsoft Macro Assembler	150	95
Microsoft Mouse *Bus Version*	175	114
Microsoft Mouse *Serial Version*	195	124
Microsoft muMath *Includes muSIMP*	300	184
Microsoft Pascal Compiler	300	184
for XENIX	695	439
Microsoft QuickBASIC	99	65
Microsoft Sort	195	129
Microsoft Windows	99	65
Microsoft Windows Development Kit	500	309

other languages

CCS MUMPS *Single-User by MGlobal*	60	51
CCS MUMPS *Single-User/Multi-Tasking* *New*	150	129
CCS MUMPS *Multi-User*	450	369
Janus/ADA C Pack *by R&R Software*	95	89
Janus/ADA D Pack *by R&R Software*	900	769
Personal REXX *by Mansfield Software*	125	99
Smalltalk/V *by Digitalk*	99	84
EGA Color Option *New*	49	45
Goodies Diskette *New*	49	45
Smalltalk/Comm	49	45
SNOBOL4+ *by Catspaw*	95	80

other products

Compact Source Print *by Aldebaran* *New*	CALL	CALL
Dan Bricklin's Demo Pgm *Software Garden*	75	59
FANSI-CONSOLE *by Hersey Micro*	75	65
FASTBACK *by 5th Generation Systems*	179	135
Informix *for DOS by Informix*	795	639
Informix4GL *for DOS by Informix*	995	789
InformixSQL *for DOS by Informix*	795	639
Instant Replay *by Nostradamus*	90	79
Interactive EASYFLOW *by Haventree*	150	129
MKS Toolkit *with vi by MKS*	139	119
Norton Commander *by Peter Norton*	75	55
On-line Help *from Opt-Tech Data Proc*	149	109
OPT-Tech Sort *by Opt-Tech Data Proc*	149	115
PrintQ *by Software Directions*	89	84
Quilt Computing Combo Package	199	159
QMake *Program Rebuild Utility*	99	84
SRMS *Software Revision Mgmt Sys*	125	109
SoftScreen/HELP *by Dialectic Systems*	195	149
Source Print *by Aldebaran Labs*	CALL	CALL
Taskview *by Sunny Hill Software*	80	56
TLIB *by Burton Systems Software*	100	89
Tree Diagrammer *by Aldebaran Labs*	CALL	CALL
VTEK *Term Emulator by Sci Endeavors*	150	129

phoenix products

Pasm86 *Macro Assembler Version 2.0*	195	115
Pdisk *Hard Disk & Backup Utility*	195	125
Pfantasy Pac *Phoenix Combo*	1295	849
Pfinish *Performance Analyzer*	395	229
Pfix-86 Plus *Symbolic Debugger*	395	229
PforCe *Comprehensive C Library*	395	229
Plink-86 Plus *Overlay Linker*	495	319
Pmaker *Make Utility*	125	78
Pmate *Macro Text Editor*	195	115
Pre-C *Lint Utility*	295	155
Ptel *Binary File Transfer Program*	195	115

polytron products

PolyBoost *The Software Accelerator*	80	69
Polytron C Beautifier	49	45
Polytron C Library I	99	78
Polytron PowerCom *Communications*	179	139
PolyLibrarian *Library Manager*	99	78
PolyLibrarian II *Library Manager*	149	115
PolyMake *UNIX-like Make Facility*	99	78
PolyShell *New*	149	119
PolyWindows Products *All Varieties*	CALL	CALL
PolyXREF *Complete Cross Ref Utility*	219	169
PolyXREF *One language only*	129	99
PVCS *Version Control System*	395	309

softcraft products

Btrieve *ISAM Mgr with No Royalties*	245	194
Xtrieve *Query Utility*	245	194
Report Option	145	114
Btrieve/N *for Networks*	595	464
Xtrieve/N	595	464
Report Option/N	345	274

text editors

Brief *from Solution Systems*	195	CALL
Epsilon *Emacs-like editor by Lugaru*	195	154
KEDIT *by Mansfield Software*	125	99
Micro/SPF *by Phaser Systems* *New*	175	139
PC/VI *by Custom Software Systems*	149	119
SPF/PC *by Command Technology Corp*	195	139
Vedit *by CompuView*	150	107
Vedit Plus *by CompuView*	185	139

turbo pascal utilities

ALICE *Interpreter by Software Channels*	95	66
Flash-up Windows *by Software Bottling*	90	79
HELP/Control *by MDS*	125	99
screenplay *all varieties by Flexus*	100	79
Screen Sculptor *by Software Bottling*	125	91
Speed Screen *by Software Bottling* *New*	125	91
System Builder *by Royal American*	100	CALL
IMPEX *Query Utility* *New*	75	CALL
Report Builder	75	CALL
TDebugPLUS *by TurboPower Software*	60	49
Turbo EXTENDER *by TurboPower Software*	85	64
Turbo Professional *by Sunny Hill*	70	48
TurboHALO *from IMSI*	129	98
TurboPower Utilities *by TurboPower*	95	78
TurboRef *by Gracon Services*	50	45
TURBOsmith *Debugger by Visual Age*	58	45
TurboWINDOW *by MetaGraphics*	80	58

wendin products

Operating System Toolbox	99	79
PCNX *Operating system*	99	79
PCVMS *Similar to VAX/VMS*	99	79
XTC *Text editor with Pascal source*	99	79

xenix system v

See also Microport System V/AT section.

XENIX System V Complete *by SCO*	1295	999
XENIX Development System	595	499
XENIX Operating Sys *Specify XT/AT*	595	499
XENIX Text Processing Package	195	144

xenix products

Btrieve *ISAM File Mgr by SoftCraft*	595	464
C-ISAM *by Informix*	319	285
C-terp *by Gimpel, Specify compiler*	498	379
c-tree *ISAM Mgr by FairCom*	395	329
dbVISTA *All Varieties by Raima*	CALL	CALL
dBx *with Library Source by Desktop AI*	550	499
DOSIX *User Version by Data Basics*	199	CALL
DOSIX *Console Version by Data Basics*	399	CALL
Informix *by Informix*	995	795
Informix4GL *by Informix*	1500	1239
InformixSQL *by Informix*	995	795
Lyrix *by Informix*	595	449
Micro Focus Level II Compact COBOL	1000	795
Forms-2	400	319
Level II ANIMATOR	600	479
Microsoft *See Microsoft Section*	CALL	CALL
Networks for XENIX *by SCO*	595	495
PANEL *Screen Designer by Roundhill*	625	535
REAL-TOOLS *Binary Version by PCT*	149	89
Library Source Version	399	289
Complete Source Version	499	369
RM/COBOL *by Ryan-McFarland*	1250	949
RM/FORTRAN *by Ryan-McFarland*	750	549
SCO Professional *Lotus clone by SCO*	795	595

LOWEST PRICES

Since this ad is prepared in advance of publication, some of our current prices may be lower than what's advertised here. Call for latest pricing.

FREE SHIPPING

Orders within the USA (including Alaska & Hawaii) are shipped FREE via UPS. Express shipping is available at the shipping carrier's standard rate with no rush fees or handling charges. To avoid delays when ordering by mail, please call first to determine the exact cost of express shipping.

CREDIT CARDS

VISA and MasterCard are accepted at no extra cost. Your card is charged when your order is shipped. Mail orders please include credit card expiration date and telephone number.

CODs AND POs

CODs and Purchase Orders are accepted at no extra cost. POs with net 30-day terms are available to qualified US accounts only.

FOREIGN ORDERS

Shipping charges for foreign and Canadian orders are based on the shipping carrier's standard rate. Since rates vary between carriers, please call or write for the exact cost. Foreign orders (except Canada), please include an additional $10 for customs form preparation. All payments must be made with US funds drawn on a US bank. Please include your telephone number when ordering by mail. Due to government regulations, we cannot ship to all countries.

VOLUME ORDERS

Call for special pricing.

SOUND ADVICE

Our knowledgeable technical staff can assist in comparing products, answer technical questions and send you detailed product information tailored to your needs.

30-DAY GUARANTEE

Most of our products come with a 30-day documentation evaluation period or 30-day return guarantee. Please note that some manufacturers restrict us from offering guarantees on their products. Call for more information.

CIRCLE NO. 175 ON READER SERVICE CARD

CALL TOLL-FREE

U S	**800-336-1166**
CANADA	**800-225-1166**
OHIO & ALASKA	
(Call Collect)	**216-877-3781**
FOREIGN	**216-877-3781**
CUSTOMER SERVICE	**216-877-1110**

Hours: Weekdays 8:30 AM to 8:00 PM EST.

Ohio customers please add 6% state sales tax.

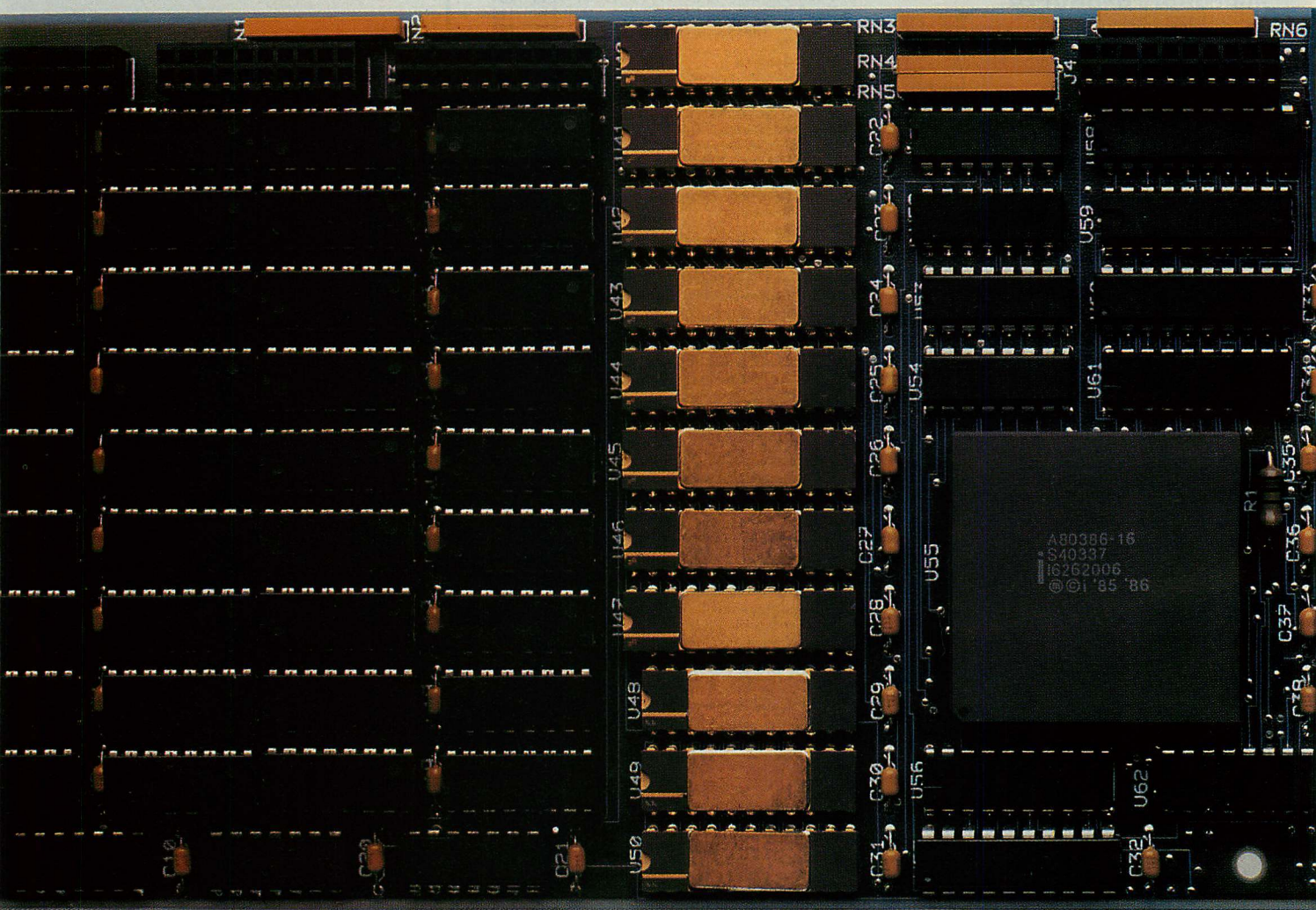
Prices are subject to change without notice.

Call or write for our FREE comprehensive price guide.

136 SUNNYSIDE STREET **HARTVILLE, OHIO 44632**

programmer's connection

How to 386



It's simple.

With Intel's Inboard™ 386/AT.

It fits right into your IBM® AT or compatible, and gives you all the performance of a 386 system.

Without having to buy a 386 system. (Which, if you've priced one lately, is about three times as expensive.)

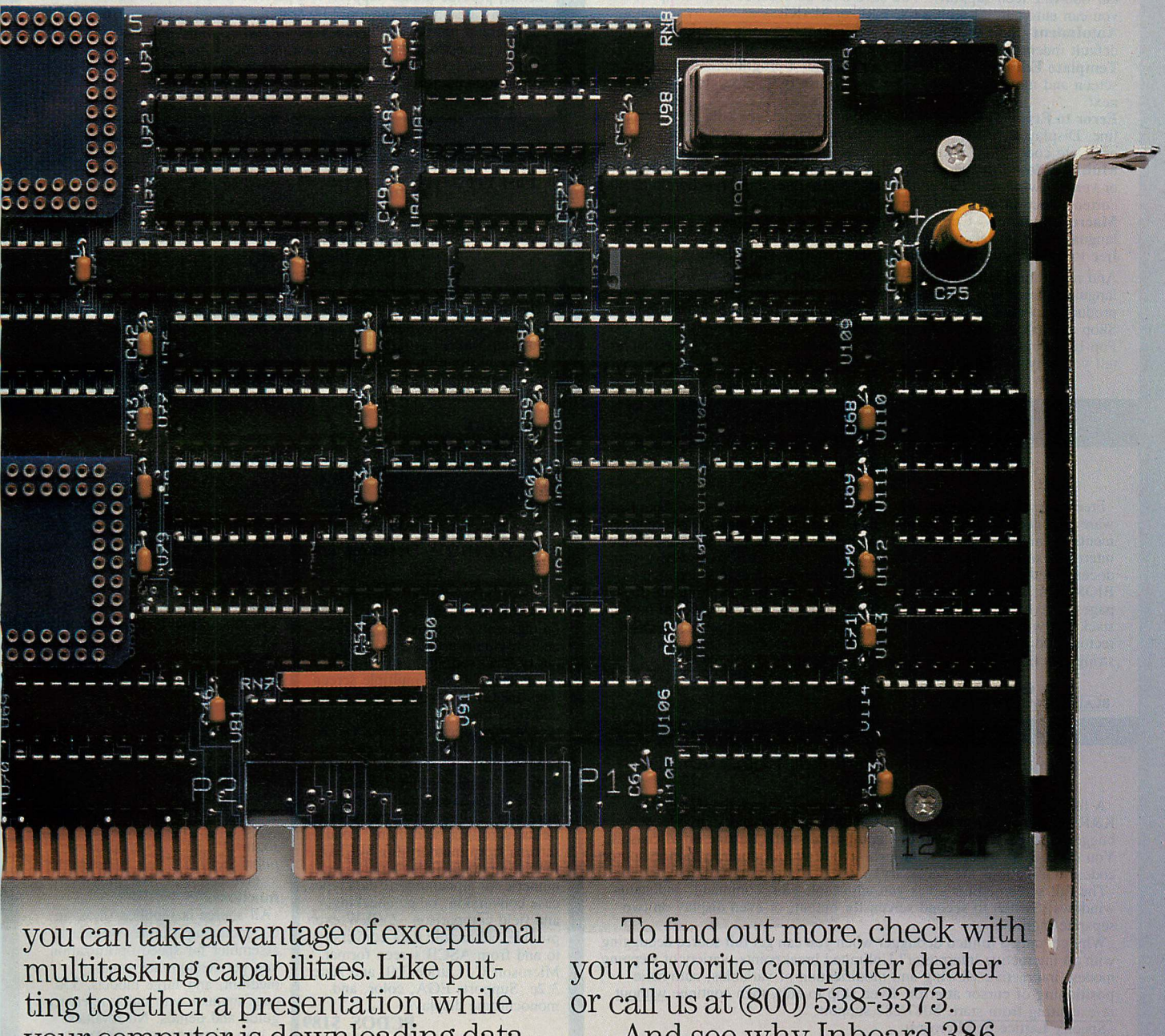
Inboard 386 is based on the revolutionary 32-bit, 16 MHz 80386 chip *we* invented. So it'll work with all the software you've got sitting on your desk. As well as any add-in boards you may have hiding in your computer—like, just for instance, the Above™ Board. Which we also invented.

Inboard 386 lets you whiz through recalcs with Lotus® 1-2-3.® And it makes your network server serve you even faster. In fact, it'll make any program serve you faster.

And with 386 control software,

Inboard and Above are trademarks and Intel a registered trademark of Intel Corporation. IBM is a registered trademark of International Business Machines Corp. Lotus and

your AT.



you can take advantage of exceptional multitasking capabilities. Like putting together a presentation while your computer is downloading data. (A slightly more efficient way of doing business.)

Don't forget our five-year warranty. Or toll-free technical support line.

To find out more, check with your favorite computer dealer or call us at (800) 538-3373.

And see why Inboard 386 beats the system.

intel®

1-2-3 are registered trademarks of Lotus Development Corp. © 1986 Intel Corporation

CIRCLE NO. 216 ON READER SERVICE CARD

C Programmers:

We support every product in this ad & 700 others.

Try any product here with a full 31-day money-back guarantee.

BRIEF Makes Editing C Programs a Breeze

No other editor comes close to making your life easier. BRIEF, The Programmer's Editor, is tailored to address the needs of C programmers — *your* needs. Take a look at the BUILT-IN features below. They are just part of the reason why 1000s of C programmers already rely on BRIEF.

AutoCompile - While in BRIEF you can compile with MS C, Lattice, or several other compilers (we'll even help you support any compiler through our 800-line tech support). You save over 20 seconds each compile; and you can automate as with a CC.EXE or MAKE.
AutoIndent - Save keystrokes and increase style consistency. Use the editor's default indentations or modify them to your taste.
Template Editing - Get the full structure of a programming construct on-screen and move the cursor from one "fill-in-the-blanks" location to the next.
Error to Error Tracking - Use "next error" to move to the next appropriate line. Display error messages in a separate window - even when you add or delete code, BRIEF knows where to go.
MultiWindow Editing - Keep different parts of the same file in horizontal or vertical windows. Put your header file in one, main function in another, current function in a third. Any size, any number.
Macro Language - Completely readable and programmable, the macro language will be second nature for an experienced C programmer. Feel free to modify the macro source code included with BRIEF.

And remember, BRIEF is a full featured editor that can be used with *any* language. User surveys indicate that *even beginning programmers* become productive with BRIEF in less than 30 minutes. Call The Programmer's Shop and ask about UNDO (not undelete), Unlimited File Size, Tiled & Pop-Up Windows . . ., or ask for a detailed product description that will tell you why BRIEF can't be beat. **800-821-2492 PCDOS $195**

NEW Blaise Tools Are Better Than Ever

C Tools Plus

Free yourself for more creative programming; stop worrying about hardware dependence. Handle everything from co-resident software requirements to multiple display pages and monitors with C Tools Plus. Filter interrupts so that other resident programs still work. 200+ well-documented functions control screen handling (direct to video adapter or BIOS calls, EGA text mode support including 43 line and multiple display pages — even handle multiple monitors), an unlimited number of pop-up, stackable windows with word-wrap, interrupt service routines, DOS directory and file handling, memory management and program control, string functions, and more. Source, no royalties. Lattice 3.0, MS C.

PCDOS $149

BLAISE COMPUTING INC.

Compiler-Compatible Interpreter, Editor, Debugger

Interactive-C™

A fully integrated development environment, Interactive-C combines a K&R standard interpreter with a full-screen editor and source-level debugger. **Interactive-C is 100% compatible with Lattice or Microsoft.** You can link in external libraries — your own or commercial: no source code modifications are necessary!

The full screen editor gives you adjustable edit, command, and status windows. Switch to second screen for output, or even display on two separate monitors.

Why get only a limited debugger when you can get full source debugging with an interactive interpreter? Unlimited breakpoints, variety of stepping modes, interactive viewing and modification of variables, automatic positioning of cursor at error. Even stop to edit, then continue without re-executing from start. 8087/287 support.

Specify Lattice or Microsoft.

PCDOS $219

MacStyle Your C Programs Use Mac's Friendly Interface with C Extender

Invention Software has written a powerful library that lets novices and professional developers alike access the Mac interface with over 85 flexible functions to extend, access, and integrate with the "Toolbox."

High and low level functions create windows, scroll bars, dialog boxes, menus, provide mouse support, and perform graphics printing with single function calls. NEW version 2.00 features ZOOM windows, marquis to copy screen image, double-click detect, search and replace multi-page printing, improved text-editing, save and recall MacPaint bit map, "bug alert," modeless dialog support, and generic I/O routines. Compact code, partial source tutorial provided, no royalties. Aztec, Megamax, Consulair, Lightspeed.

MAC $119

Fast Prototyping and Development of User Interfaces with Skylights

"It's much easier to adapt to the end-user's needs than any other product I've seen . . . definitely a programmer's tool."
- William Elswick, Software Engineer, Compact Video

Quickly design interactive prototypes, then include screens you develop in your finished application code. Design demos or tutorials.

Skylight combines an intuitive screen/window/menu editor, run-time windowing, menu handling, and front-end support routines, and "Demo/Tutorial Maker" program plus detailed low-level primitives. Supports a variety of pointing devices (mice, tablets, lightpens). All major C compilers; even use with other languages (BASIC, Pascal, Assembly) with utility included Bit-mapped graphics upgrade available. No royalties.

PCDOS $359

Skylight Software, Inc.

Flexible SCREEN and WINDOW Development with

ZVIEW Screen Library

Use this field-sensitive tool to develop data entry screens and windows and provide run-time flexibility. Security level settings restrict inquiry or update of fields; multiple screen help display is available at screen and field level.

NEW Features: Windows can be stacked, peeled off, and moved at run-time. You also get automatic scrolling of data within fields.

ZVIEW gives you full control of attributes, colors, boxes, protected fields, scrolling, and more. Load screens from memory for SPEED. Field support includes alpha, numeric, or alphanumeric data types, case conversion, range checking, and field comparison. ZVIEW even provides automatic data conversion to and from ASCII screen format. Microsoft C, Lattice 3.0, and Aztec 3.2e. Supports EGA, color, and monochrome displays.

PCDOS $189

280 Functions Without "Fat":

Blackstar 'C' Function Library

Without duplicating compiler library functions, the Blackstar 'C' Function Library covers the range. BOTH Microsoft C and Lattice C library versions are included in one product with screen, graphics, file, text, and serial communications routines. Extensive device driver support (including mouse handling). Low-level utilities and DOS and memory control routines. And a great price, too.

Some other libraries are padded with different functions that do the same thing, or functions that repeat routines supplied with your compiler. Sterling Castle trims the fat to give you **over 280 unique functions.**

All source is supplied (most in C), including routines written in assembler for speed optimization. You get versions for small, medium, and large models. 350 page manual with quick reference guide and demos. No royalties.

1-800-7-CASTLE **PCDOS $99**

Call for Your FREE C Programmer's Directory

VISA HOURS MasterCard

8:30 AM - 8:00 PM EST.

CIRCLE 134 ON READER SERVICE CARD

CIRCLE NO. 220 ON READER SERVICE CARD

800-421-8006

THE PROGRAMMER'S SHOP™
5-P Pond Park Road, Hingham, MA 02043
Mass.: 800-442-8070 or 617-826-7531 11/86

"I appreciate your service to the programming community, your prices more than fair and your newsletter amongst the finest in the business."
Lawrence
Fooian Systems, Inc.

TECH NOTEBOOK 74

DEAN P. GIENGER

XON/XOFF Printer Driver

By using the XON/XOFF software handshaking method shown here, the PC can support peripherals that have serial interfaces.

Although the IBM PC supports parallel printers with ease, there are times when driving serial peripherals would be desirable. Many popular printers, such as the Hewlett-Packard LaserJet, use serial (or RS-232) interfaces.

Because most printers can print characters at only a fraction of the rate that a PC can send them, a printer must exchange some *handshaking*, or control signals, to control the rate at which characters are sent by the PC. This control can be done through either hardware or software.

The hardware method involves connecting a special wire between the printer and the PC. When the printer is ready to accept data, it asserts a logic 1 on this wire. When the printer is unable to accept data, it asserts a logic 0. The PC must then check the status of this signal before sending data. This simple method of flow control has several drawbacks. First, it requires an additional wire in the interface cable. Second, the printer cannot return any additional status data (paper out, device off-line, and so on) unless additional wires are connected between the PC and the printer.

The software method involves having the printer send special flow control characters to the PC to notify it of status conditions. When the printer is able to accept data it will send a special character, called an XON character—chr(17). When the printer is unable to accept data, an XOFF character—chr(19)—is sent by the printer to the PC. Other characters may be sent to notify the PC of special conditions.

The PC may be configured with either serial or parallel ports; however, the PC's software drivers support only the hardware method of handshaking. The program below provides a software driver that allows the PC to use a serial printer that uses XON/XOFF software handshaking.

The program "steals" the special BIOS printer interrupt vector reserved for printer functions. This vector is located in low memory at address 0000:005CH and is normally set to point to the printer driver code located in the system ROM. The program redirects this vector to point to its own code for driving the serial port using XON/XOFF handshaking. Once the XON program is loaded, it directs all printer output to the serial port. A serial printer then can be used as if it is a standard parallel interface printer.

Dean P. Gienger is an engineering consultant in California.

LISTING 1: XON.ASM

```
; Use DOS MODE to redirect LPT1: to COM1: and set up the BAUD rate etc
rom_ptr_vec   equ     17h*4           ; rom vector location
com           equ     03f8h           ; com1=3f8, com2=2f8
code          segment public 'CODE'
              assume  cs:code,ds:code
              org     0100h
start         proc    far             ; patch in the new code
              cli                     ; disable interrupts
              sub     ax,ax           ; set es segment
              mov     es,ax           ; copy vector
              mov     ax,cs           ; reset ptr vector
              mov     es:word ptr[rom_ptr_vec+2],ax
              mov     ax,offset ptr_int
              mov     es:word ptr[rom_ptr_vec],ax
              sti                     ; reenable interrupts
              mov     dx,offset last_byte  ; call dos to exit
              int     27h             ; and stay resident
              db      '(c) 1986, Dean P Gienger'
start         endp
ptrint        proc    far             ; we get here from int 17h
ptr_int:      sti                     ; enable interrupts
              cmp     ah,0            ; is this an output req?
              je      ptr_out         ; yes
              cmp     ah,1            ; is it an init req?
              je      ptr_init        ; yes
ptr_stat:     push    dx              ; it must be a status req
              mov     dx,com+5        ; get status
              in      al,dx
              and     al,20h
              je      ptr_busy        ; port is busy
              mov     dx,com
              in      al,dx           ; is there an xoff ?
              and     al,7fh
              cmp     al,19           ; xoff is ^S ( 19 )
              je      ptr_busy        ; yes, it is busy
              mov     ah,80h          ; else return not busy
              pop     dx
              iret
ptr_busy:     mov     ah,00h          ; return busy status
              pop     dx
              iret
ptr_out:      push    ax              ; print character & save
ptr_wait:     mov     ah,2            ; wait till printer free
              int     17h             ; ready?
              and     ah,80h
              jz      ptr_wait        ; no
              pop     ax              ; yes, get char from stk
              push    dx
              mov     dx,com          ; send character
              out     dx,al
              pop     dx
              mov     ah,2            ; return status
              int     17h
              iret
ptr_init:     mov     ax,27           ; initialize the printer
              int     17h             ; send ESC E
              mov     ax,45h          ; to init printer
              int     17h
              mov     ah,2            ; return status
              int     17h
              iret
last_byte:
ptrint        endp
code          ends
              end
```

COMPATIBILITY 386 PERFORMANCE

COMPAQ DESKPRO 386

The New Standard

Compaq has forged ahead of IBM with the first 80386-based machine, offering AT compatibility with twice the performance—and a bright future in multitasking.

STEVEN ARMBRUST and TED FORGERON

Until now, Compaq Computer Corporation has been content with matching IBM's personal computer offerings, merely making minor improvements to what IBM already had created. With the introduction of the Deskpro 386, however, Compaq has leapfrogged over IBM into brand new territory. Powered by a 16-MHz Intel 80386 microprocessor and equipped with 32-bit memory and a speedy hard-disk drive, the Deskpro 386 not only offers compatibility with IBM's PC/AT at twice the performance, but also promises to run future multitasking operating environments that will take advantage of the 80386 processor.

The Deskpro 386 comes in three models, distinguished only by the hard disk. The model 40 is equipped with a 40MB hard disk, the model 70 with a 70MB hard disk, and the model 130 with a 130MB hard disk. The monitor, the amount of memory, and the keyboard are options that can be selected with each model. The accompanying sidebar lists the features available with the Deskpro 386. The unit reviewed in this article was the model 40 with 1MB of RAM, a 1.2MB diskette drive, a 360KB diskette drive, a 40MB tape backup unit, a serial and a parallel port, the Compaq Enhanced Color Graphics Board, and the Compaq Color Monitor.

At first glance, the Deskpro 386 looks just like the other members of the Deskpro family. It uses the same cabinet as the Deskpro 286 and therefore has many of the same advantages and disadvantages as that unit has (see "Out from the Shadow of IBM: Compaq Deskpro 286," Steven Armbrust and Ted Forgeron, August, 1986, p. 80). The size of the system unit is 19¾ inches by 16½ inches by 6¼ inches. Photo 1 compares its footprint with that of an AT.

Like the Deskpro 286, the Deskpro 386 includes the same unattractive key-lock switch on the front of the system unit, with barely readable switch positions. It also has the on/off switch posi-

PHOTOGRAPH • MARC DAVID COHEN

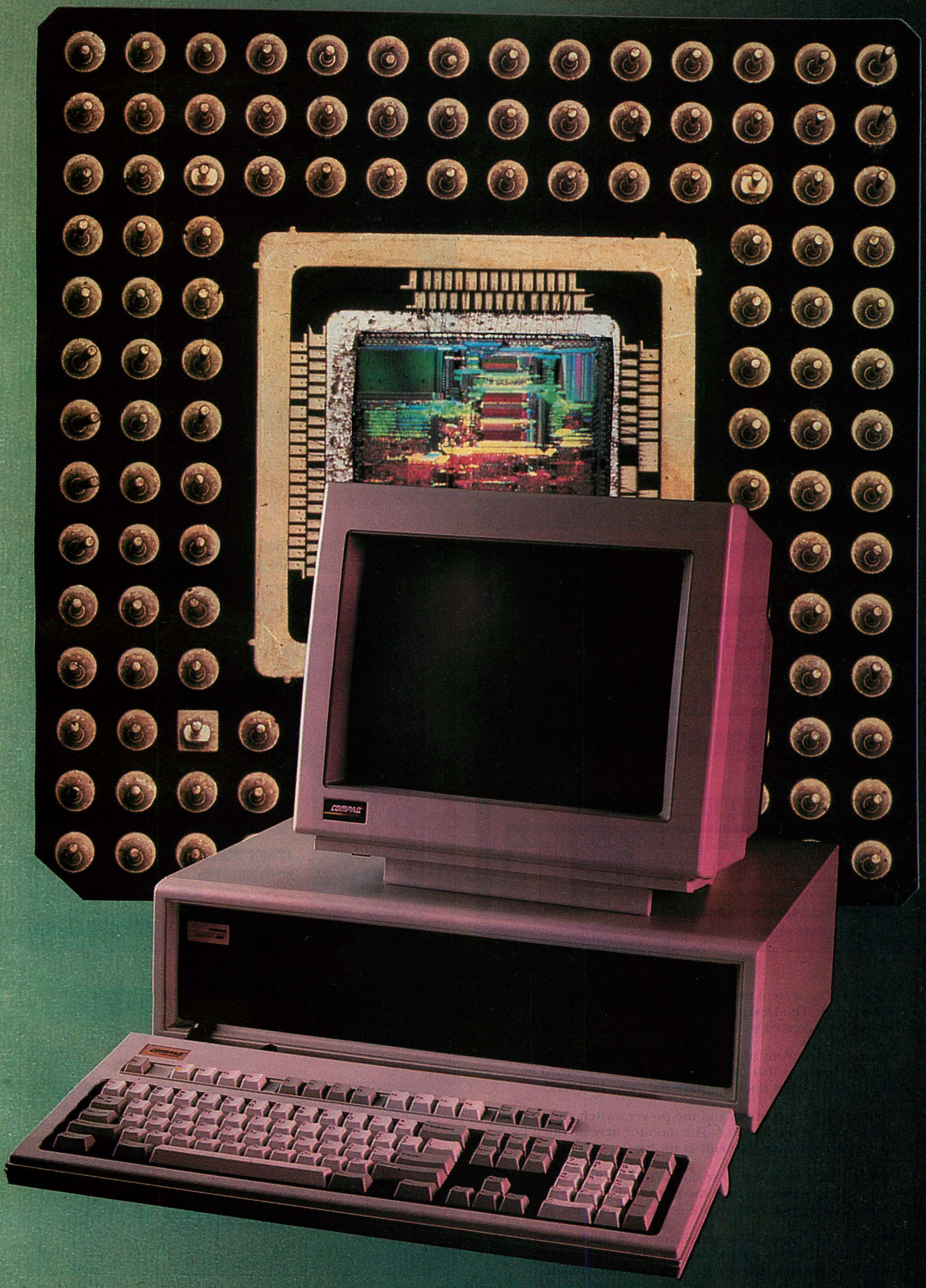

COMPAQ

PHOTO 1: *System Unit Footprint*

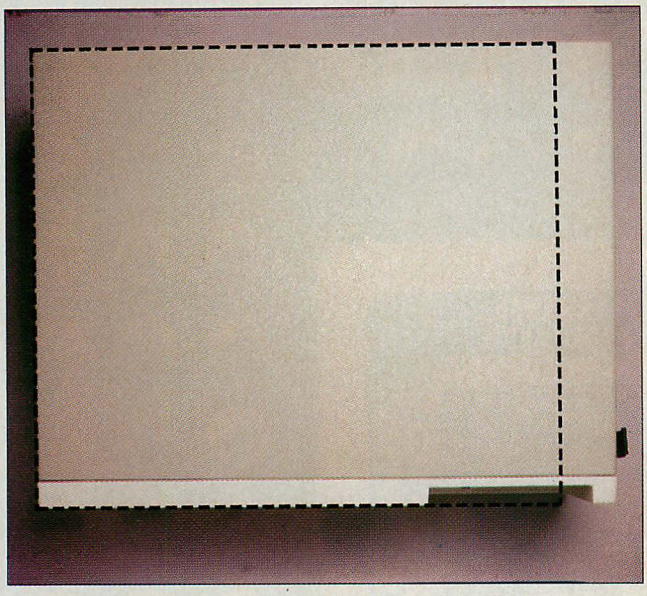


PHOTO 2: *Keyboard Comparison*

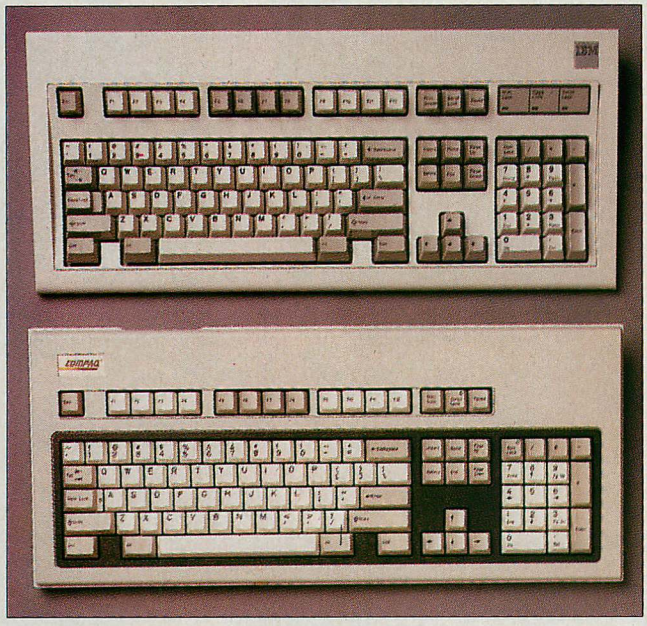


PHOTO 3: *Compaq Color Monitor*

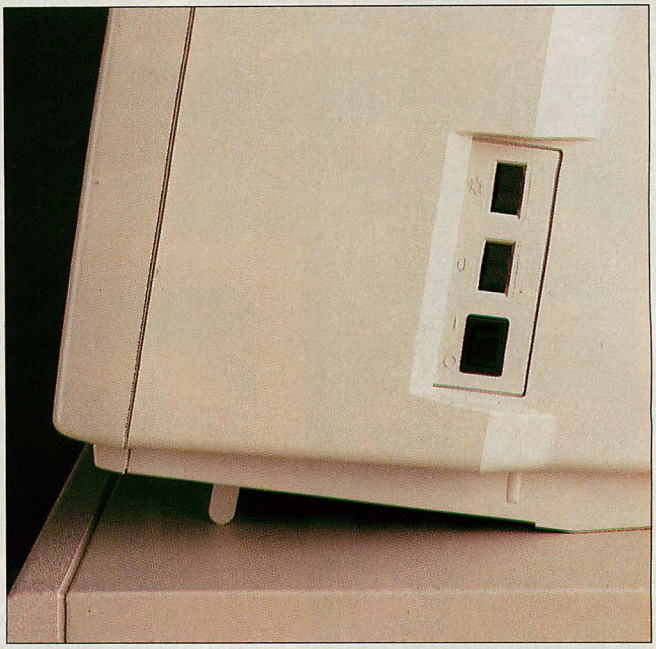


PHOTO 4: *Inside the System Unit*

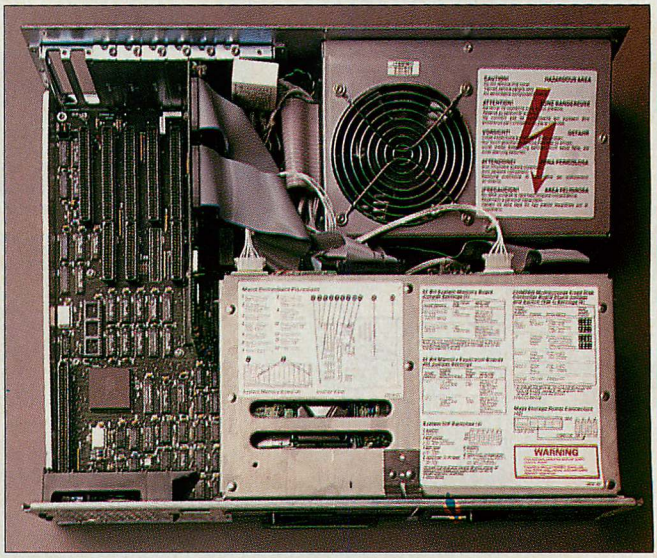


PHOTO 5: *System Memory Board*

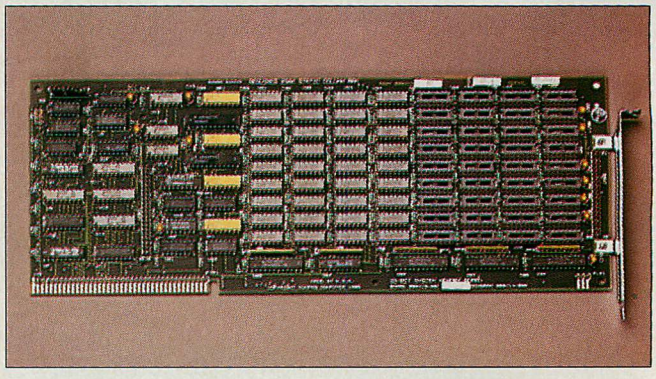


Photo 1: The system unit of the Deskpro 386 is smaller than the IBM AT. The Deskpro measures 19¾ by 16½ by 6¼ inches, as indicated by the dotted lines.

Photo 2: The layout of the enhanced keyboard of the Deskpro 386 matches that of the IBM enhanced keyboard in most respects, but does not have the same IBM feel.

Photo 3: Screen controls and the power switch are located on the side of the monitor. The monitor may be tilted up 5 or 10 degrees using the built-in tilt bar.

Photo 4: Slot 1, the leftmost slot, is the 32-bit slot. The main connector for this slot is at the front of the system unit. A small power/ground connector is located near the rear.

Photo 5: The main 32-bit connector is on the bottom left of the board. The two double rows of pins protruding from the board are for attaching a piggyback memory module.

tioned on the rear panel of the unit where it is difficult to find. The two machines share the convenient feature of a two-color drive light that shines green when accessing a 1.2MB diskette and red when accessing a 360KB diskette.

Two keyboards are available for the Deskpro 386: one matching the layout of IBM's original AT keyboard and one matching IBM's enhanced keyboard. The only difference between the new Compaq keyboard and the earlier one is in the layout. The keys still have a light touch and provide little tactile feedback. The keyboard plugs into the front of the system unit, requiring its removal whenever the cover of the system unit is taken off. As with other Deskpro models, the keyboard cable seems to be excessively long, but this problem can be alleviated by tilting up the keyboard on its legs and placing the excess cable underneath. Photo 2 compares Compaq's enhanced keyboard with that of IBM.

With the Deskpro 386, Compaq offers its new Color Monitor and Enhanced Graphics Board, which is compatible with the IBM Enhanced Graphics Adapter (EGA). The Compaq Color Monitor is compatible with IBM's Enhanced Color Display. It produces accurate colors and text that is easy to read. However, when compared to the new generation of enhanced color monitors, such as the NEC Multisync, the characters produced by the Compaq monitor in high-resolution text mode are not as crisp and clear. The monitor controls are conveniently placed on the side of the display where they can be accessed easily (see photo 3).

The Compaq graphics board works with both the color display and Compaq's dual-mode monochrome display. When connected to the dual-mode monochrome display and switched into monochrome emulation mode, the Compaq graphics board can produce text with a resolution of 720-by-350 pixels. When connected to the Compaq Color Monitor (or another enhanced color display), it offers 16 colors from a palette of 64 and a resolution of 640-by-350 pixels. The graphics board also can be used with an ordinary color display to generate output equivalent to that produced by IBM's Color Graphics Adapter (CGA). However, it cannot be connected to an IBM monochrome monitor (or the equivalent) without damaging the display. The board can be installed in other Compaq computers, such as the Portable or Portable II, where it can drive either the built-in dual-mode display or a color display.

80386 ENHANCEMENT

The Deskpro 386 is loaded with extras that are not available on any other AT-compatible computer, but by far the most exciting and obvious of these is the 80386. This 32-bit microprocessor from Intel has 32-bit instruction and data paths, provides real and protected modes, and offers a virtual-8086 mode and memory paging that enable it to handle multiple real-mode applications (such as DOS applications) simultaneously. (For a complete description of the capabilities of the 80386, see "Upward to the 80386," Caldwell Crosswy and Mike Perez, February 1987, p. 50.)

The Deskpro 386 system starts running with the 80386 in real mode, and a reboot always returns the processor to this mode. When operating in real mode, all the restrictions associated with 8086/88 processors apply. Segments are limited to 64KB, and the memory address space is 1MB. The 80386's 32-bit instructions can be used, however, enabling programs to improve performance with 32-bit data transfers and 32-bit operands for instructions.

Protected mode in the Deskpro 386 is compatible with the protected mode in 80286 machines, but the 80386 increases the memory address space (from 16MB to 4GB) and the maximum segment size (from 64KB to 4GB). In addition, it also provides memory paging, I/O protection, a full 32-bit instruction set, and virtual-8086 mode.

Virtual-8086 mode is a special form of protected mode that enables real-mode applications to execute within protected mode. In virtual-8086 mode, memory addressing reverts to the base:offset form used in the 8088 and in real mode on the 80286 and 80386. This permits current application programs to run, but confines them to the 1MB address space of the 8088. Despite this limitation, the paging facilities (described below) of the 80386 permit the 1MB real-mode address space to be mapped anywhere in the 4GB address space that is available in protected mode, thus permitting several real-mode applications to run concurrently—without any modification.

The Deskpro 386 uses this virtual-8086 mode and memory paging combination to implement its own Compaq Expanded Memory Manager (CEMM). This program gives DOS applications access to the expanded memory outlined in the Lotus-Intel-Microsoft (LIM) specification without requiring special bank-switching memory boards.

In the Deskpro 386, the 80386 processor runs at 16 MHz, twice the speed at which the 80286 runs in the AT. In addition, the 80386 can access RAM 32 bits at a time (compared with the 16-bit memory access provided by the 80286). With this double dose of supercharging, DOS applications might be expected to run four times as fast as they do on an 8-MHz AT; however, because the cur-

DESKPRO 386 VITAL STATISTICS

Model 40: $6,499
1MB memory
Parallel printer interface
Serial interface
1.2MB diskette drive
Realtime clock
40MB hard disk

Model 70: $7,299
All features of Model 40 except with a 70MB hard disk.

Model 130: $8,799
All features of Model 40 except with a 130MB hard disk.

Display adapters (none is standard)
Dual-mode
Enhanced Color Graphics Board

Monitors (none is standard)
Dual-mode monochrome monitor
Enhanced color monitor

Memory capacity on system board
None

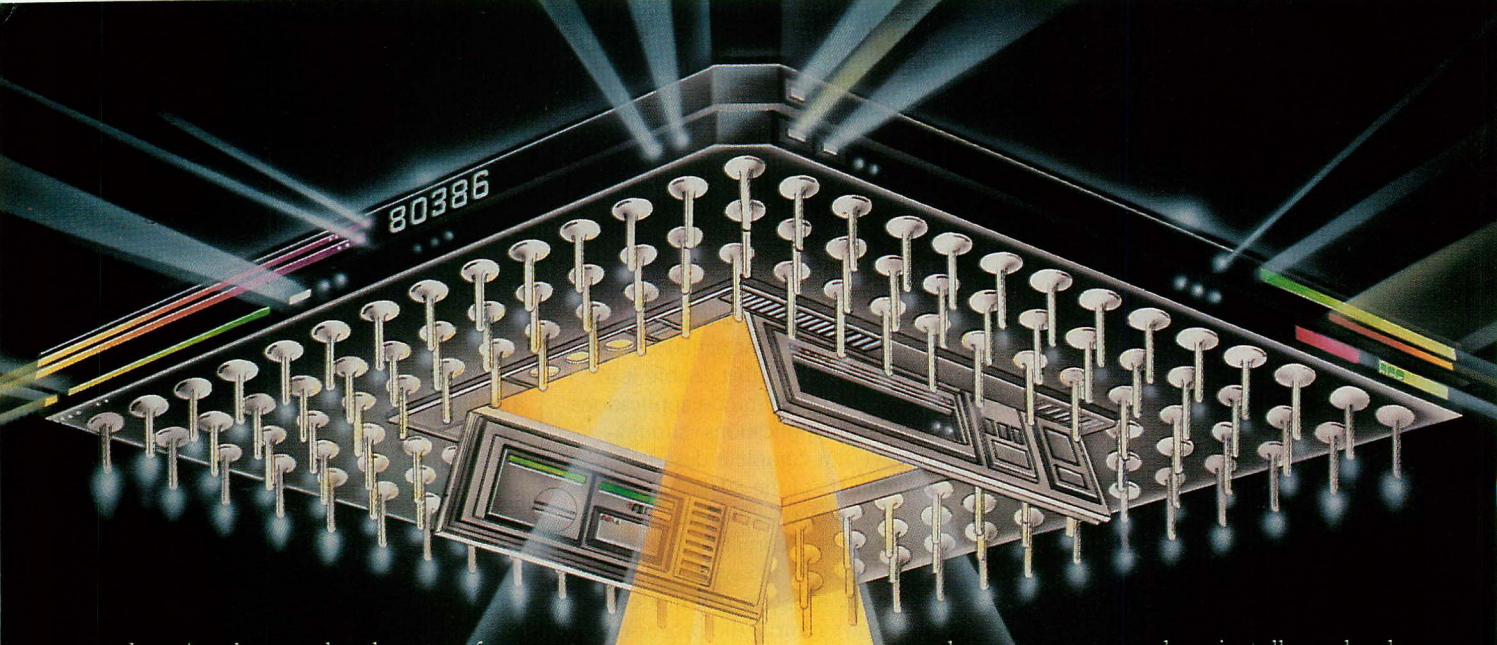
32-bit memory capacity of system
10MB

Expansion slots
32-bit: 1
16-bit: 4
8-bit: 3 (one half-size slot)

Available slots

Model 40		Model 130	
32-bit:	0	32-bit:	0
16-bit:	3	16-bit:	2
8-bit:	2	8-bit:	2

Other extras available
40MB tape backup: $799
360KB diskette drive: $225
1MB memory option: $549
1MB/2MB expansion board: $849
4MB/8MB expansion board: $2,995



Imagine the speed and power of a $100,000 minicomputer in a desktop PC costing under $7,000. Now imagine all that power going to waste because the operating system you chose was never meant to take advantage of a computer this powerful. It will take more than just a "window environment" or an outdated operating system to unlock the 80386.

It will take PC-MOS/386™.

The First 80386 Operating System. Specifically designed for the 80386 computer, PC-MOS/386™ opens doors. Doors to more memory and multi-tasking. Doors to thousands of DOS programs as well as upcoming 80386-specific software. It's the *gateway* to the latest technology..., and your networking future.

Memory Management Without Boards. PC-MOS exploits the memory management capabilities built into the 80386. So, up to four GIGABYTES of memory are accessible to multiple users and to future 80386-specific applications requiring megabytes of memory.

Multi-Tasking, Multi-User Support for One, Five or 25 Users. PC-MOS/386™ allows up to 25 inexpensive terminals to be driven by a single 80386 machine. So the features of the 80386 can be utilized at every terminal. And it comes in three versions so you can upgrade your system as your company grows...without having to learn new commands or install new hardware.

Software Support for Thousands of DOS Programs. Although PC-MOS/386™ totally replaces DOS, it doesn't make you replace your favorite DOS programs. So you can run programs like Lotus 1-2-3, WordStar, dBASE III, and WordPerfect on the 80386. Best of all, it uses familiar commands like DIR and COPY–so you'll feel comfortable with our system.

The Gateway to Endless Features. Distinctive characteristics like file/system security, remote access, file/record locking, and built-in color graphics support for EACH user set PC-MOS/386™ apart from all previous operating systems.

Open the Doors to Your Future TODAY! Call The Software Link TODAY for more information and the authorized dealer nearest you. PC-MOS/386™ comes in single, five & 25-user versions starting at $195.

UP TO 25 USERS.

MADE FOR THE 80386.

RUNS DOS PROGRAMS.

MULTI-TASKING



THE SOFTWARE LINK, INC.

Developers of LANLink™ & MultiLink® Advanced

8601 Dunwoody Place, Suite 632
Atlanta, GA 30338 Telex 4996147 SWLINK

CALL: 800/451-LINK

In Georgia: 404/998-0019
Dealer/OEM Inquiries Invited
Dealers: 404/998-0700 OEMs: 404/641-8554

THE SOFTWARE LINK, INC./CANADA
CALL: 416/477-5480

More Than Just Windows, We've Opened Doors.

TRADEMARK ACKNOWLEDGEMENTS: MultiLink® is a registered trademark of The Software Link, Inc. PC-MOS/386™ MultiLink® Advanced, and LANLink™ are trademarks of The Software Link, Inc. Lotus 1-2-3, WordStar, dBASE III, & WordPerfect are trademarks of Lotus Development Corp., MicroPro, Ashton-Tate, & WordPerfect Corp., respectively. Prices and technical specifications subject to change.

CIRCLE NO. 196 ON READER SERVICE CARD

rent versions of DOS and the applications that run under them are written for 16-bit processors, code and data are still accessed 16 bits, rather than 32, at a time. Despite this underutilization of the 80386, existing real-mode programs run approximately twice as fast on the Deskpro 386 as on an 8-MHz AT.

In the future, programs that take advantage of the 32-bit instruction set of the 80386 can expect to improve performance even more. For example, the 32-bit version of the MOV instruction, which is available even in real mode, can potentially double the data-transfer speed provided by the 80286.

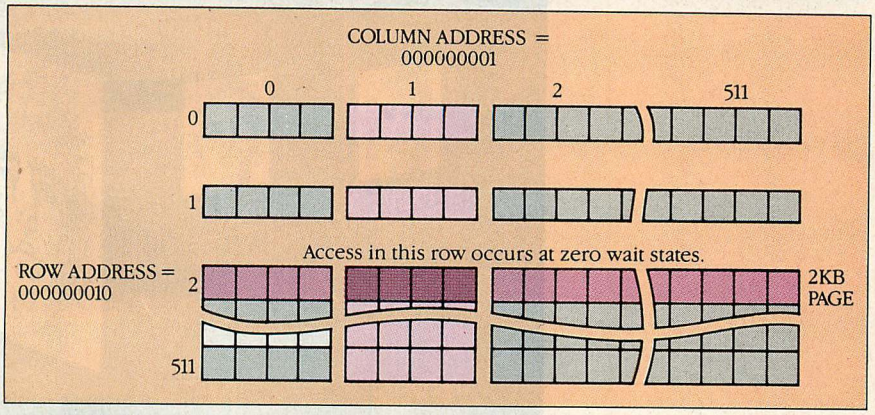
To take advantage of the 32-bit data path of the 80386, the Deskpro 386 provides a 32-bit memory bus for access to RAM. This bus takes the form of a 32-bit slot on the system board. Compaq claims that this is not a general-purpose bus, but merely a mechanism to optimize the performance of the Deskpro 386 memory subsystem. It enforces this opinion by providing just a single 32-bit slot and by not supplying any RAM on the system board itself. A 32-bit memory board, supplied in the standard configuration with 1MB of soldered RAM, fits into this slot. With additional 256K-by-1 static-column DRAM chips, this board can be expanded to 2MB. One of two piggyback boards can be added to the 32-bit memory board: one provides 1MB, expandable to 2MB, with 256K-by-1 chips; the other provides 4MB, expandable to 8MB, with 1M-by-1 chips. Thus, the Deskpro 386 can contain as much as 10MB of RAM, all accessed via the single 32-bit slot.

The 32-bit slot consists of an 80-pin connector whose signals are generated directly by the 80386. This memory bus, which runs at 16 MHz, is simply an extension of the processor's local bus. This direct interface minimizes the signal delays between the processor and the memory subsystem.

The 32-bit memory boards are configured into a 36-bit wide arrangement, consisting of 32 bits of data and 4 bits of parity (one parity bit for each byte). On the system memory board, and on the 1MB/2MB piggyback board, the memory chips are 256K-by-1 static-column DRAMs, yielding a memory bank size of 256K 32-bit double words, or 1MB. On the 4MB/8MB piggyback board, the memory chips are 1M-by-1 DRAMs, yielding memory banks of 4MB.

With a 16-MHz processor, fast access to RAM is essential; otherwise, the speed advantages of the processor are negated. In the PC, the cycle time of the 8088 is 210 nanoseconds (ns), and a bus cycle is 840 ns, enabling readily available 200-ns DRAM chips to be used. The 200 ns is the access time, or the time required for a charged DRAM to return information. Additional time is required to recharge the DRAM for the next memory access. The sum of the access time and the recharge time is the DRAM cycle time, which is the actual time needed to obtain information from the chip. In 200-ns DRAM chips, the cycle time is 345 ns.

FIGURE 1: *RAS/CAS Memory Access*



To access memory in each 1MB bank, the memory subsystem specifies nine-bit row and column addresses. Static-column RAM allows the memory to respond in zero wait states if the row address is the same as in the previous memory access.

The cycle time of a 16-MHz 80386 is 62.5 ns. Using 200-ns RAM chips and an ordinary memory-accessing scheme would reduce the overall performance to little better than that of a PC, because wait states would have to be inserted to make the processor spend some of its time waiting for memory.

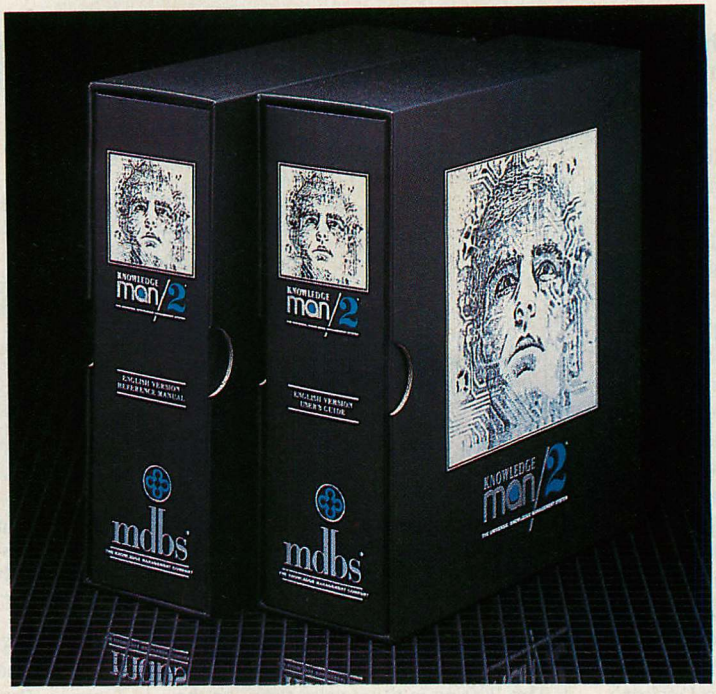
The easiest way to improve memory access times is to remove all wait states and use faster DRAM chips. However, to make significant improvements, chips with memory access times in the 60-ns range are required, and those are prohibitively expensive. Computer manufacturers have resorted to other schemes to obtain better performance from the memory subsystems on high-performance computers while still using less expensive DRAM chips.

One method is to use a memory cache, a limited amount of high-speed static RAM that is used to store copies of the memory locations that are accessed most often. With a large enough memory cache and an appropriate algorithm for copying data to and from the cache, many caching systems are 90-percent effective (that is, more than 90 percent of the time, the data needed will already be in the high-speed cache). Caches are more expensive than some other methods, because they require costly, high-speed SRAM chips in addition to the RAM in the user-address space. Extra board space is also required for the memory that makes up the cache and its support circuitry.

Another method for improving memory access times uses interleaved (or bank-switched) memory. This scheme assumes that most memory accesses are sequentially ordered. Therefore, the memory is divided into two or more banks, with the sequential addresses interleaved among the banks. For example, in a 32-bit computer with two memory banks, the first 32 bits would be in bank 0, the second 32 bits in bank 1, the third 32 bits back in bank 0, and so on. This interleaving speeds sequential accesses by enabling one memory bank to be fetching a subsequent double word while the processor accesses the previous double word from the other bank. However, the benefit of overlapped accesses is not without penalty. Additional logic is required to implement the interleaved memory, and initial accesses (and consecutive accesses to the same bank) are longer than they would be if the interleaving circuitry were not present.

Interleaving improves the performance of sequential accesses, but real mode on the 80386 (which almost all current applications use) requires a segmented program structure—that is, separate (often physically separated) segments are used to store a program's code and data. As a result, memory accesses often bounce around between the code and data segments with few extended periods of sequential accesses; thus, much of the advantage of an interleaved architecture is lost.

A third alternative, and the one used in the Deskpro 386, is memory



KNOWLEDGEMAN/2 LETS YOU GET TODAY'S WORK DONE AND PLAN FOR TOMORROW.

KnowledgeMan/2,® the most advanced business software available, lets you be more productive than ever before and in less time.



These powerful business capabilities, tightly fused into a single program, work together for you:

☐ Relational data management ☐ Spur-of-the-moment inquiry ☐ Spreadsheets ☐ Statistical analysis ☐ Forms management ☐ Programming language

☐ Options for creating graphs, processing text, generating reports, painting forms, mouse processing and remote communications.

KnowledgeMan/2's four user interfaces make it easy for you...and all your business people...to work with KnowledgeMan/2:

☐ Menus, help screens and easy-to-use documentation guide your every step, if you're just beginning.

☐ Direct commands for the power user, with help available when you want it.

☐ K-Chat, the optional natural language interface, for asking for information in plain English.

☐ Procedural programming for the advanced user.

Get KnowledgeMan/2 working for you today. And start planning for tomorrow.

KnowledgeMan/2 is available for the IBM PC, PC/XT, PC/AT and compatibles. Multi-user versions are offered for the DEC VAX system and 3Com, Novell and IBM PC local area networks.

Your local **mdbs** sales representative can provide additional information. For the location of the sales office nearest you, call or write **mdbs.**

mdbs and KnowledgeMan/2 are registered trademarks and K-Chat a trademark of **mdbs**, Inc. "VAX" is a registered trademark of Digital Equipment; "PC/XT" and "PC/AT" of IBM Corp.; 3Com of 3Com Corp.; Novell of Novell, Inc.

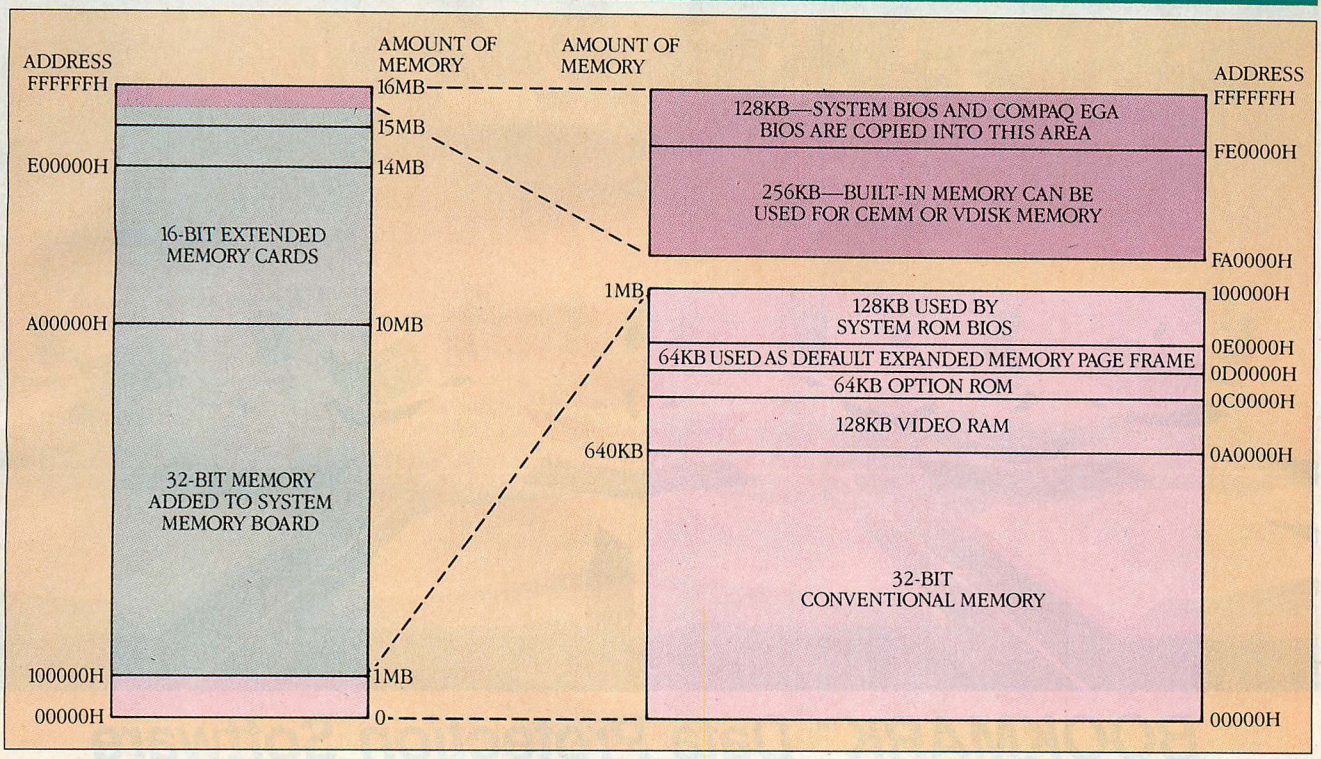
KNOWLEDGE man/2®
THE UNIVERSAL KNOWLEDGE MANAGEMENT SYSTEM

mdbs/Marketing and Sales · P.O. Box 248 · Lafayette, IN 47902 · 800/344-5832 · 317/463-2581 (Indiana, Canada)

mdbs®

CIRCLE NO. 211 ON READER SERVICE CARD

FIGURE 2: *Deskpro 386 Memory Map*



When the Deskpro 386 is set up for 640KB of conventional memory, the remaining memory in the first megabyte is located at addresses below 16MB. The BIOS is copied into the top 128KB. Other 32-bit memory is assigned addresses beginning at 1MB.

paging, in which the system RAM is divided into areas of equal size called pages. With the paging circuitry in operation, successive accesses to memory within a page are extremely fast, while successive accesses to different pages are slower. The segmented architecture of real mode negates some of the advantages of paging, just as it does with interleaving. Accessing memory outside a page boundary enacts the same type of performance penalty as consecutively accessing a single bank of an interleaved system.

The Deskpro 386 implements memory paging by using static-column RAM chips. For memory access, each 1MB bank of RAM is logically divided into rows and columns, and access to memory is obtained by presenting a row address and column address. These addresses are latched internally using the row address strobe (RAS) and column address strobe (CAS). The RAS and CAS combination identifies a specific 32-bit double word—the amount of memory that the 80386 can access in a single memory fetch. Each row and column address is a 9-bit quantity, implying 512 (or 2^9) items in each row and column. With each item being 4 bytes (32 bits) long, each page (or specific row address) of memory consists of 512 (2^9 column addresses) by 4 bytes, or 2KB of memory. With 512 of these pages, the addressing scheme is able to handle 1MB of memory. Additional 1MB banks are accessed in the same manner, with their own row and column addresses (see figure 1).

With static-column DRAM chips, memory accesses within the same page can happen with zero wait states, because the RAS is maintained from the previous memory access. When a new memory access is required, the row address is compared with the value already maintained by the memory chips. If the new row address is the same as the previous one (that is, the memory is in the same 2KB page), only a new column address needs to be presented to the DRAM. This results in a zero-wait-state access. However, if the new row address is different (that is, the memory is in a different 2KB bank), the processor must wait for the memory to recharge and present a new row address followed by column address. This process results in two wait states.

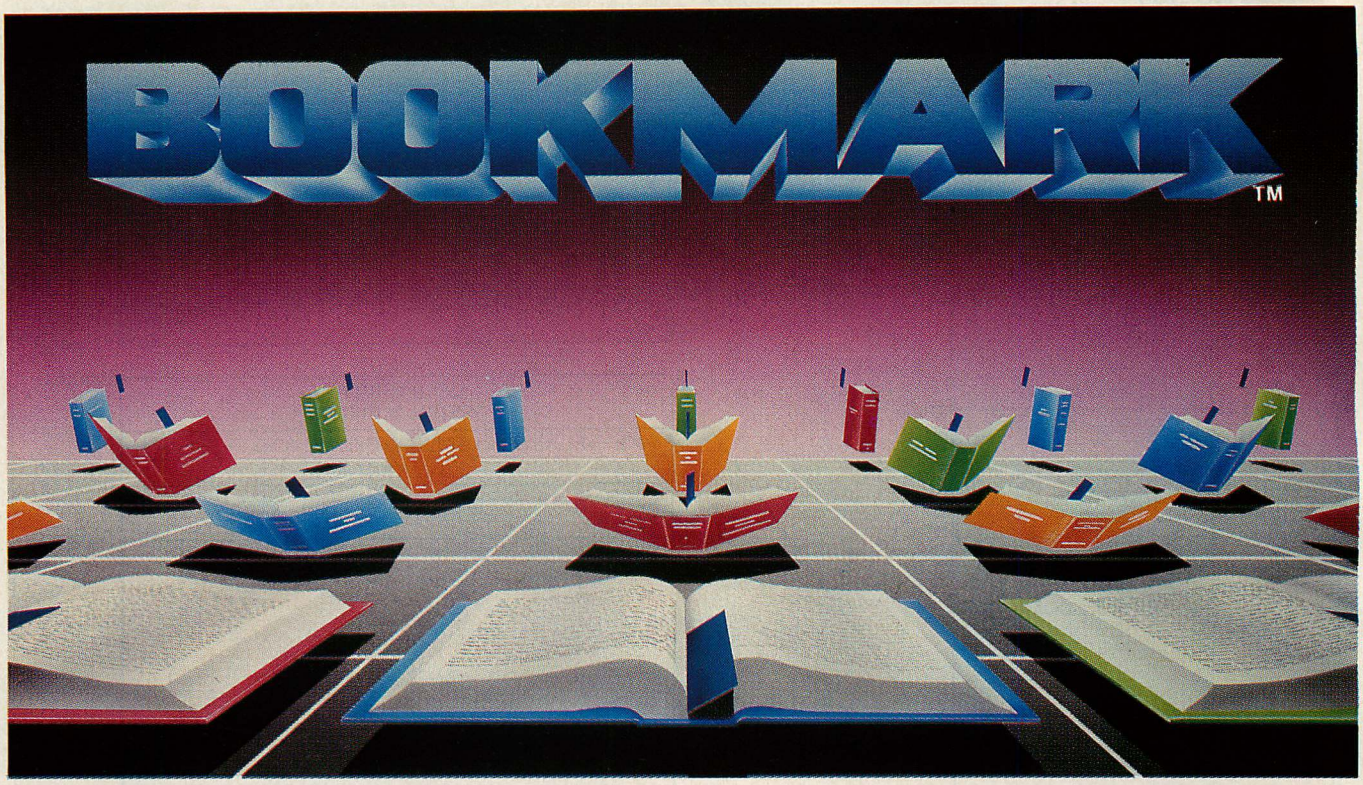
There is one situation, however, in which access to memory occurs at two wait states, even if the access is within the same page. If an idle bus cycle occurs, the memory subsystem turns off paging mode, allowing the memory subsystem to get a head start on whatever memory access it predicts will happen next. If the next access is within the same page, however, it will occur at two rather than zero wait states, because the processor will have to restart paging mode and present both a row and a column address.

Idle bus cycles are common when writing to memory because the number of processor cycles required to execute some write instructions is more than the bus cycles needed to transfer the data. Thus, idle bus cycles are inserted, which turns off the paging.

Compaq turns off paging when an idle cycle occurs because tests showed that an idle cycle typically precedes an access outside the page. Therefore, even though some accesses within a page are slowed to two wait states, overall performance is increased by 1 to 2 percent by using the idle cycle to anticipate accesses outside a page.

The paging scheme used in the Deskpro 386 complements the way that the 80386 instruction prefetch works. The sequential nature of instructions and the 80386's desire to keep the prefetch queue full enable the memory subsystem to operate within a physical page much of the time.

Compaq claims that in average circumstances, approximately 60 percent of all memory accesses occur within a page. Therefore, the memory in the



BOOKMARK™ Data Protection Software

Automatically marks your place so you can resume where you left off.

Runs with most software so if you decide to call it a day you can power down, and upon restart BOOKMARK will bring you back where you left off...it's like having a BOOKMARK in your computer!

Stop wasting time navigating through menus . . . or teaching novices.

Easy to use. You choose how often work is saved, by number of keystrokes or length of time.

Work in progress can be resumed within seconds after system crash, power failure, accidental reset or power down.

Costly battery backup systems are no longer needed when work in progress is already saved to the point of the previous BOOKMARK placement.

BOOKMARK is a memory-resident utility that automatically and periodically saves work in progress to a reserved area of the hard disk.

Suggested Retail Price **$69.95**

Requirements:
IBM PC/XT/AT or 100% Compatible
64k to 640k
IBM DOS 2.1 or Higher
1 Floppy Disk Drive • Hard Disk Drive (10 Megbytes Minimum) • Video Display Adapter, IBM (Monochrome, Color, Enhanced Color), Hercules, AST
(BOOKMARK occupies an equivalent space on hard disk as in system RAM plus video RAM)

To Order, Call INTELLISOFT International or Ask Your Dealer for BOOKMARK

INTELLISOFT™

INTERNATIONAL

Call Toll Free (800) 544-MARK · In California Call Toll Free (800) 543-MARK
70 Digital Drive · P.O. Box 1972 · Novato, California 94948 · (415) 883-1188

BOOKMARK™ is a trademark of INTELLISOFT International. Copyright © 1986 by INTELLISOFT International. All rights reserved.
IBM PC/XT/AT are registered trademarks of International Business Machines Corporation. Hercules is a trademark of Hercules Computer Technology.
AST is a registered trademark of AST Research, Inc.

CIRCLE NO. 114 ON READER SERVICE CARD

Deskpro 386 must average 0.8 wait states per 32-bit access. For tightly coded, highly sequential applications, such as graphics drivers, it is possible to approach zero wait states.

To put these figures into perspective, consider the difference in access times between 32-bit memory on the Deskpro 386 and 8-bit memory over the PC bus. A 32-bit access within the same page on the Deskpro 386 requires 2 processor cycles (at 62.5 ns per cycle) or 125 ns. A 32-bit access outside the current page requires 4 processor cycles or 250 ns. Assuming 60 percent of the accesses occur within the current page, the average 32-bit access time is 175 ns. A similar 32-bit access from 8-bit memory requires 48 processor cycles, or approximately 3 microseconds. No wonder the Deskpro 386 seems fast.

Tests conducted for this article revealed that it was possible to achieve zero wait states when reading and writing memory. However, normal programs will not achieve this level of performance unless their memory accesses are limited to the same 2KB page. One wait state is more realistic.

It remains to be seen whether the memory paging scheme used in the Deskpro 386 really performs better than interleaving or is more cost-effective than caching. Once the various kinds of systems become readily available, more conclusive real-world performance measurements can be taken.

NO MEMORY WASTED

In its base configuration, the Deskpro 386 contains 1MB of 32-bit RAM soldered into a system memory card that fits into the 32-bit expansion slot. With jumpers on this memory card, the user can select 256KB, 512KB, or 640KB of conventional memory. The remainder of the 1MB is automatically mapped to just beneath the 16MB address. Figure 2 shows the memory map for the most common configuration (640KB).

Although the extra 384KB of RAM is not available via normal DOS operations, Compaq has not let that extra memory go to waste. The uppermost 128KB segment, located at address FE0000H, is used to improve the BIOS performance. Upon start-up, the Deskpro 386 copies the system BIOS and the Compaq Enhanced Graphics Board BIOS (if available) into that memory and remaps the memory via special hardware map registers so that it replaces the 128KB area normally reserved for the system ROM (from 0E0000H to 0FFFFFH). Once this mapping takes place, applications can access the remapped BIOS either at address FE0000H or 0E0000H; the actual system and graphics board ROM devices are no longer accessible. Even with the remapping, this procedure relocates the graphics board BIOS, which normally starts at 0C0000H, to 0E0000H to 0F0000H. Therefore, only the BIOS from a Compaq Enhanced Graphics Board is copied because there is no assurance that another BIOS will contain position-independent code that works properly when relocated.

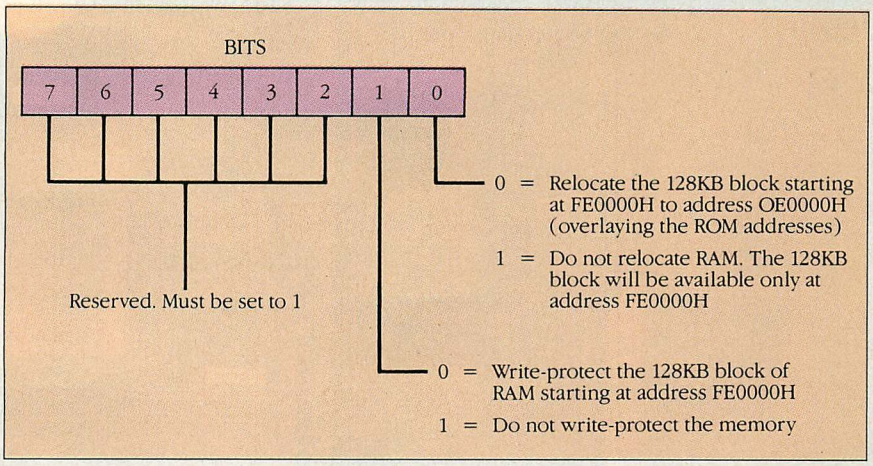
By remapping the BIOS into 32-bit RAM, the Deskpro 386 dramatically improves the BIOS performance. Instead of residing on 250-ns ROM chips accessed via a 16-bit bus in the case of the system BIOS or via an 8-bit bus in the case of the enhanced graphics BIOS, the remapped BIOS is accessed out of 32-bit static-column RAM. This RAM is write protected by special hardware registers so that out-of-control programs do not accidentally write to the BIOS and cause damage.

The Deskpro 386 contains a memory-mapped hardware register called the RAM relocation register that controls the mapping and write protection of this system RAM. It is a write-only register available at address 80C00000H. Bits 0 and 1 control the remapping of the ROM space and the write protection, as shown in figure 3.

Reading the same memory location (80C00000H) returns the contents of the diagnostics register. This register can pinpoint which byte within a 32-bit double word caused a parity error. It also identifies which memory is 32 bits so that diagnostics programs can use 32-bit reads and writes to speed up their RAM tests. Figure 4 shows the bit definitions of the diagnostics register.

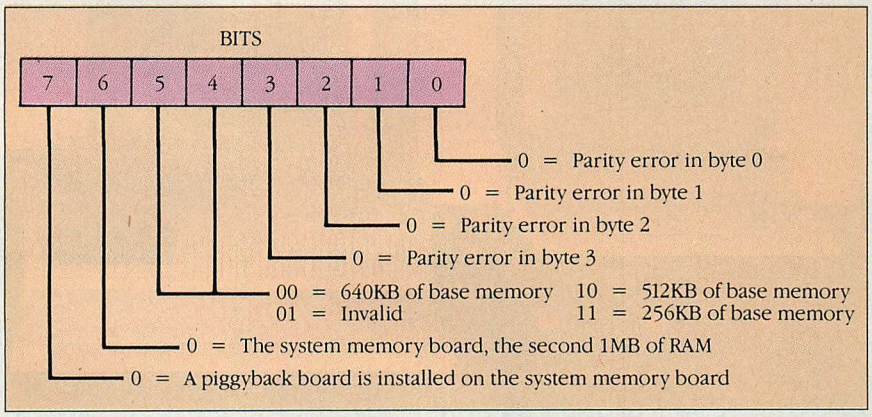
Despite the speed improvements caused by the remapped ROM, this feature is not likely to improve overall

FIGURE 3: *Memory-mapped Hardware Register*



Writing to location 80C00000H sets the memory-mapped hardware register that is used to remap and write-protect the memory into which the ROM BIOS is copied.

FIGURE 4: *Memory-mapped Diagnostics Register*



Reading location 80C00000H returns the status of the memory configuration jumpers and the parity bits for each of the four bytes of a 32-bit double word.

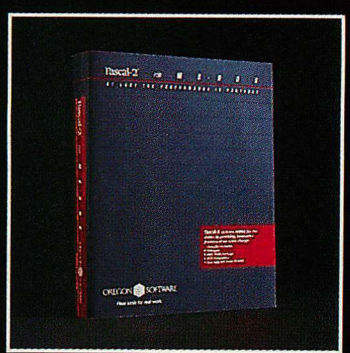


TALK OF THE TOWN

One language supports this community. That language is Pascal-2, now on the PC and producing the fastest, most compact code available. For the professional programmer, imagine what you can do with this power: ■ Cut execution time by 20% to 200% ■ Transport MS-DOS programs to VAX, PDP11, and 68000 machines with only minor adjustments ■ Cut executable program size by up to 50% ■ Use all of DOS-addressable memory through efficient large-memory model ■ Speed error correction and save development turn-around time with sophisticated error checking and reporting ■ Find and fix logical errors with the interactive source-level debugger ■ Access DOS services and network files ■ Call Microsoft FORTRAN, C, Pascal, and assembler ■ Upgrade from TURBO Pascal with compatible strings, equivalent procedures and access to TURBO graphics.

Pascal-2™

FOR MS-DOS



Plus!

■ *Intel CEL87 mathematical library for scientific computing* ■ *A special interface between Pascal-2 and the programmable BRIEF text editor (editor optional).* ■ *Certified ISO standard Level 1.*

Dramatically improve your productivity and introduce your PC software to the VAX next door.

Call or write OREGON SOFTWARE, INC.
6915 SW Macadam Avenue,
Portland, OR 97219 (800) 367-2202
TWX: 910-464-4779 FAX: (503) 245-8449

OREGON SOFTWARE

Real tools for real work

AT LAST THE PERFORMANCE IS PORTABLE

The following are trademarks: Oregon Software, Pascal-2, Oregon Software, Inc.; IBM, PC-AT, PC-DOS International Business Machines Corporation; Intel, Intel Corporation; MS, Microsoft Corp.; TURBO Pascal, Borland International, Inc.; BRIEF, UnderWare Corp.; PDP, VAX, Digital Equipment Corp.

CIRCLE NO. 200 ON READER SERVICE CARD

performance appreciably. The most performance-sensitive areas of the BIOS are the screen display routines, and most applications circumvent those routines by writing directly to the display adapter RAM via the 8-bit interface. Programs that use the BIOS for screen operations, such as DIR, TYPE, and other DOS commands, are much faster. Thus, the remapped BIOS can affect users' perceptions of overall performance, simply because many people judge a computer's speed by how fast a DIR listing scrolls up the screen.

On the other hand, if the BIOS were not copied into RAM, the performance of some tight loops that use the BIOS might actually be worse than on an 80286-based computer. This degradation could occur because the 80386 is optimized for 32-bit operations, while the BIOS is contained on ROM chips accessed via the 16-bit bus.

For example, the 80386 is constantly trying to fill its prefetch queue with the instructions it expects will execute next. This prefetch queue is 32 bits wide, so each prefetch is 32 bits, whether or not the bits come in one access from 32-bit memory, two accesses from 16-bit memory, or four accesses from 8-bit memory. A prefetch of system BIOS instructions from ROM requires two separate 16-bit accesses, both of which must be completed before the processor can continue with other operations. Because of the extra time required for two separate memory accesses, a processor that was idle at the start of the prefetch might be holding up other operations by the time it finishes. This effect will be especially noticeable in programs that have small (two instruction) loops, because the jump instruction at the end of the loop flushes the prefetch queue. Users also might notice this effect if they do not use the Compaq Enhanced Graphics Board, because the Deskpro 386 will not copy non-Compaq BIOS code into RAM. In this case, four separate 8-bit accesses are required for each 32-bit fetch because the graphics board uses an 8-bit interface.

The remaining 256KB of RAM not available via normal DOS operations is referred to by Compaq as *built-in memory*. Two programs are provided to enable ordinary DOS applications to take advantage of this memory: CEMM and VDISK, a RAM-disk program. Other programs will have difficulty using this built-in memory, because it resides at high memory addresses just below the remapped BIOS. Most programs that use extended memory require that such memory begin at the 1MB address and that no gaps exist in the memory. This is not a problem, however, for the CEMM and VDISK programs. Even if other extended memory is added, CEMM and VDISK can use both the built-in memory and the other extended memory automatically.

CEMM was developed jointly with Microsoft to provide an expanded memory interface to extended memory. CEMM uses the virtual-8086 mode of the 80386 in combination with the 80386's paging capability to support the LIM expanded memory specification, without the need for special bank-switching memory boards such as Intel's Above Board. CEMM simulates the bank-switching hardware by mapping the Deskpro 386's 32-bit extended memory into the page frame specified in the LIM standard. This enables DOS applications that support the LIM specification to access additional memory up to the 8MB limit of the specification.

CEMM grants flexibility to the large amount of fast 32-bit memory that can be added to the Deskpro 386. For example, Intel's QUIKMEM2 RAM disk, which normally requires an expanded memory board, ran without difficulty from extended memory when using CEMM. The added flexibility has its cost, however. System overhead for handling interrupts increases in virtual-8086 mode because of the time required for the 80386 to switch between protected and virtual-8086 modes. Thus, programs that run in virtual-8086 mode execute approximately 5 percent slower than programs that run in real mode.

Two programs are provided to enable ordinary DOS applications to take advantage of built-in memory: CEMM and VDISK.

In addition, other programs that run in protected mode cannot be used when CEMM is active. Attempts to run such programs (RAM disks and disk caches are two examples) cause CEMM to display the following message:

Privileged Operation Error
Deactivate CEMM and Continue (C) or reBoot (B) (C/B)?

When this message occurs, the user can either deactivate the expanded memory and continue with the program or reboot the computer.

To avoid such messages when running software that uses protected mode, the user can switch expanded memory on or off from the keyboard, using the commands CEMM ON and CEMM OFF. CEMM AUTO is the default setting, making expanded memory (and its associated overhead) available when requested by an application program.

386 EXTRAS

As mentioned earlier, additional memory and piggyback cards can be added to the Deskpro 386 system memory card to supply as much as 10MB of high-speed, 32-bit RAM. Other 16-bit memory cards such as those used in the AT can be installed in addition to or in place of some of the 32-bit RAM. However, the performance difference between the 16-bit expansion bus and the 32-bit memory bus is so extreme that 16-bit memory should be added only as a last resort.

Another convenient feature of the Deskpro 386 is its processor speed control, which is designed to provide high performance while maintaining compatibility with programs that depend on program execution speed for critical timing functions. With the SPEED option of the MODE command (or BIOS interrupt 16H, AH=F0H), the processor speed can be set anywhere from slightly less than the 4.7 MHz used in the PC's 8088 to 16 MHz. Unlike the Deskpro 286, the Deskpro 386 does not support special keyboard sequences to switch in or out of any of these modes (the Deskpro 286 uses Ctrl-Alt-/ to change speeds). Only the MODE command or BIOS interrupt 16H can be used. Five options of the MODE command affect the processor speed:

MODE SPEED = COMMON
MODE SPEED = FAST
MODE SPEED = HIGH
MODE SPEED = AUTO
MODE SPEED = num

The COMMON setting causes the 80386 to simulate the speed of a 6-MHz 80286. FAST simulates an 8-MHz 80286. HIGH sets the speed to a full 16 MHz.

In AUTO mode, the default speed setting, the processor operates at full 16-MHz speed until a diskette access occurs; then it switches to 8-MHz (FAST) mode until the disk access is complete, when it switches back to 16 MHz. The BIOS recognizes when to switch speeds by keying on the diskette's motor-on switch. The AUTO mode is intended to help users install and run copy-pro-

tected programs, particularly those that require a key diskette. As Compaq realized, most copy-protection schemes are speed sensitive. With the processor slowed just during diskette accesses, the copy-protection mechanism can succeed without forcing the computer to run permanently at the slower speed. After installation (or key-diskette checking) is complete, the Deskpro 386 returns to 16-MHz operation.

The last form of the MODE SPEED command allows users to specify a numeric value from 1 to 50 for the processor speed. This form of the command corresponds to the way the speed is set using BIOS interrupt 16H and permits users to select from a large variety of speeds. Table 1 shows how these numbers correspond with the other MODE SPEED options.

The Deskpro 386 implements the speed controls by using the 80386 HOLD signal. The 80386 is in a HOLD state whenever a memory refresh occurs. Therefore, to slow down the processor, the refresh period is simply extended, reducing the amount of time that the 80386 is actually executing instructions. This method of slowing down the processor is better than adding memory wait states because it does not affect the bus bandwidth or the latency for direct memory access (or other bus master operations). However, extremely tight program loops that execute within the time of a refresh cycle and require a slowed processor might not work correctly.

The Deskpro 386 uses a signal from an 8254 interval timer to lengthen the refresh period. The 8254 timer is programmed using the values in the BIOS interrupt 16H; function request AH=F0H; the MODE SPEED=num command; or the values that correspond to the other MODE SPEED options.

Normally, the processor speed is set in its default state (AUTO). This permits 16-MHz speed during most system operations, but still allows copy-protected programs to be loaded from diskette. The other settings can be useful for running speed-sensitive programs such as games. One especially nice touch is that any selected speed survives a warm reboot of the computer. This means that even action games written especially for a 4.77-MHz 8088 can be run by first selecting a low speed, such as MODE SPEED=03 and then rebooting the computer with the game disk in drive A:.

Because the Deskpro 386 needs an 8254 signal for its speed control, it contains two 8254 timers, one more than

TABLE 1: *MODE SPEED Parameters*

SYSTEM SPEED SIMULATED	MODE COMMANDS
8088-based PC	MODE SPEED=3
6-MHz AT	MODE SPEED=COMMON or MODE SPEED=16
8-MHz AT	MODE SPEED=FAST or MODE SPEED=18
16-MHz 80386	MODE SPEED=HIGH or MODE SPEED=50

The MODE SPEED command can be used to change operating speed. This table correlates the MODE SPEED parameters with the numeric values entered.

the AT. Both timers have three separate counters. In the first timer, which provides the same functions as the 8254 on the AT, counter 0 is connected to the 8259 interrupt controller and provides the system timer interrupt for time-of-day, diskette time-out, and other system functions; counter 1 generates the refresh request signal; and counter 2 generates the tone for the speaker.

The second 8254 timer provides functions not found in the AT. Counter 1 of the second timer is unused; counter 2 extends the refresh period as just described; and counter 0 provides one of the most interesting features of the machine. It is a fail-safe timer that can generate interrupts on the nonmaskable interrupt line at regular intervals. The Compaq documentation states that operating systems can use this signal to prevent the system from locking up. It also claims that this function is intended for future operating system use. Therefore, this system "heartbeat" might be included in versions of DOS developed for the 80386 processor. A switch on the system board can turn the fail-safe timer on or off.

One feature that the Deskpro 386 might be expected to contain, but does not, is a socket for an 80387 numeric coprocessor.

A Motorola MC146818 serves as the Deskpro 386's realtime clock and stores configuration information. This component, which also is used in the AT, contains 64 bytes of memory. The first 14 are used by the clock, and the last 50 are used to save the system configuration when the power is turned off. Three of the bits in configuration byte 45 (2DH) store additional information not maintained by the AT: bit 0 indicates whether Compaq's dual-mode monitor is installed; bit 1 indicates whether the audible key-click feature is enabled; and bit 3 indicates whether a Compaq or non-Compaq graphics adapter is installed.

The Deskpro 386 has other features not found in the AT, most of which are also available in the Deskpro 286. These include the Ctrl-Alt-plus and Ctrl-Alt-minus sequences to increase or reduce the loudness of the key clicks and the optional tape backup unit.

The speed and capacity of the tape backup unit has been improved over the units previously available with the Deskpro computers. Tape backup now is twice as fast as earlier models; and the new unit can back up 40MB of information instead of 10MB, because it uses the 3M DC2000 3½-inch cartridges rather than the DC1000 cartridges. For compatibility with earlier Compaq computers, the new tape unit can read the DC1000 tapes produced by other Compaq tape units, but it cannot write to them. The new tape unit can be installed in the Deskpro 286.

Unfortunately, on several occasions the tape drive in the tested unit refused to back up data successfully. It locked up during the middle of a backup operation and could not be restarted except by turning the power off and on again. The backup operation was tried at several processor speeds, but the errors still occurred frequently. These problems did not reappear when another unit was tested, but the reliability of Compaq's higher-capacity tape drives should be questioned nonetheless.

One feature that the Deskpro 386 might be expected to contain, but does not, is a socket for an 80387 numeric coprocessor. This omission probably occurred because the design of the 80387 was incomplete when the Deskpro 386 was developed. The machine does have a socket for the 80287, which has a different number of pins and a different pin layout than the 80387. A further drawback is that the Deskpro 386 supports only 4- and 8-MHz 80287s, not the 10-MHz parts that are now

widely available. A switch on the system board indicates whether a 4- or 8-MHz 80287 is installed.

LOOKING INSIDE THE 386

The box housing the system unit of the Deskpro 386 is virtually identical to that of the Deskpro 286, so the joys and sorrows of installation are also the same. On the negative side, Torx screws are used in most of the fastenings, and Torx screwdrivers are often difficult to find. Further, a metal brace is mounted on the left side of the system unit, immediately over slot one (the 32-bit slot). To take out the memory board, the brace has to be removed.

On the positive side, the cover of the system unit is held on by fewer screws on the rear panel so it is easier to remove than the cover of the Deskpro 286. Instead of selecting the correct three of eight screws, all of which look as if they need to come out, users now choose three of four screws.

Compaq has improved the installation process immensely by gluing a quick-reference card to the top of the peripheral bay enclosure. This card contains a drawing of the system board with callouts identifying all the expansion cards, the 80287 socket, the system board switches, the power supply and power connectors, and the system ROMs. In addition, it identifies the drive connectors, lists the settings for the system board switch, and lists the jumper selections for the memory board and disk controller board. With this convenient reference, the experienced user can perform almost all installation tasks without opening a manual.

The peripheral bay in the Deskpro 386 is capable of containing four half-height drives. This is a smaller area than the AT provides, but the extra space underneath the bay is filled with shock-mounting material to protect the drives from damage.

One well-designed feature is the easy-to-reach location of the 80386 and 80287 sockets on the left side of the system board. The B1 stepping of the 80386 is used in the Deskpro 386. Although this version of the 80386 has several programming anomalies, Compaq's technical reference contains 10 pages of valuable information that describes how to avoid problems with the stepping of the 80386. Most of the anomalies are of interest only to designers of protected-mode operating systems. Application programmers will rarely encounter any problems.

As shown in photo 4, the Deskpro 386 has eight expansion slots—three 8-bit slots (one of them half size), four 16-bit slots, and one 32-bit slot. The 8-bit and 16-bit slots operate at 8 MHz for compatibility with boards designed to operate in the AT. The 32-bit slot is slot 1, which is the leftmost slot in the photo. This slot consists of two connectors: an 80-pin connector positioned at the front of the system unit and a small one next to slot 2 that is actually a second set of power and ground signals for improved power distribution.

The system memory board, shown in photo 5, fits into the 32-bit slot. The board comes standard with 1MB of RAM, which is soldered in place. Sockets are available for an additional megabyte on the board, as are connectors for adding a single piggyback card. The memory board and piggyback card accept only static-column RAM chips. With the largest piggyback card installed, 10MB of RAM can be accessed from the system memory board.

The system memory board comes standard with 1MB of RAM, which is expandable to 2MB. More can be added using a piggyback card.

The system board has only one switch block. It specifies the monitor type, default speed setting, whether the numeric coprocessor is present and, if so, the speed at which it runs, and whether the fail-safe timer is enabled.

Of the remaining seven slots, two are occupied with a 40MB hard disk—one 8-bit slot by the graphics controller and one 16-bit slot by the multipurpose hard-disk controller board. The multipurpose board controls the 40MB hard disk, the diskette drives, and the tape drive. On the units with 70MB or 130MB hard disks, another 16-bit slot is occupied, because the larger capacity drives require a separate controller.

In the model 40 unit, the multipurpose hard-disk controller is connected to two diskette drives (a 1.2MB drive and a 360KB drive), a 40MB hard disk, and a 40MB tape backup unit. The two diskette drives and the tape backup unit slots are attached to a single connector on the controller in a daisy-chain fashion. The controller supports a second 40MB drive, but the power supply, the same 192-watt model used in the Deskpro 286, is not powerful enough to handle two hard disks and a tape drive. Therefore, if a second hard disk is added, the tape backup unit must not be included. To pack all of that functionality into a single controller board, Compaq uses special 40MB disks. The drives contain much of the control circuitry on the drive circuit board rather than on the controller board.

The 70MB and 130MB drives do not connect to the multipurpose controller board. Both of these drives use the ESDI interface, so they require a separate controller card that supports a single drive. Both controllers use the same I/O addresses when communicating with the hard disks, so if a 70MB or 130MB drive is included, no 40MB drives are supported.

Although only the 40MB, 70MB, and 130MB drive options are available from Compaq, the ROM BIOS supports 47 types of hard disks. Table 2 lists the drive types and characteristics. All the Deskpro 386 drive types are supported by the AT, with the same type numbers.

A serial and a parallel port are also contained on the multipurpose hard-disk controller. The serial port uses a 9-pin, male, D-shell connector. The parallel port uses a 25-pin, female, D-shell connector. These connectors are the same as those used in the AT's parallel/serial card. The serial port can be switched between COM1 and COM2. The parallel port is set to LPT1, but it can be disabled if it conflicts with the ports on other expansion cards.

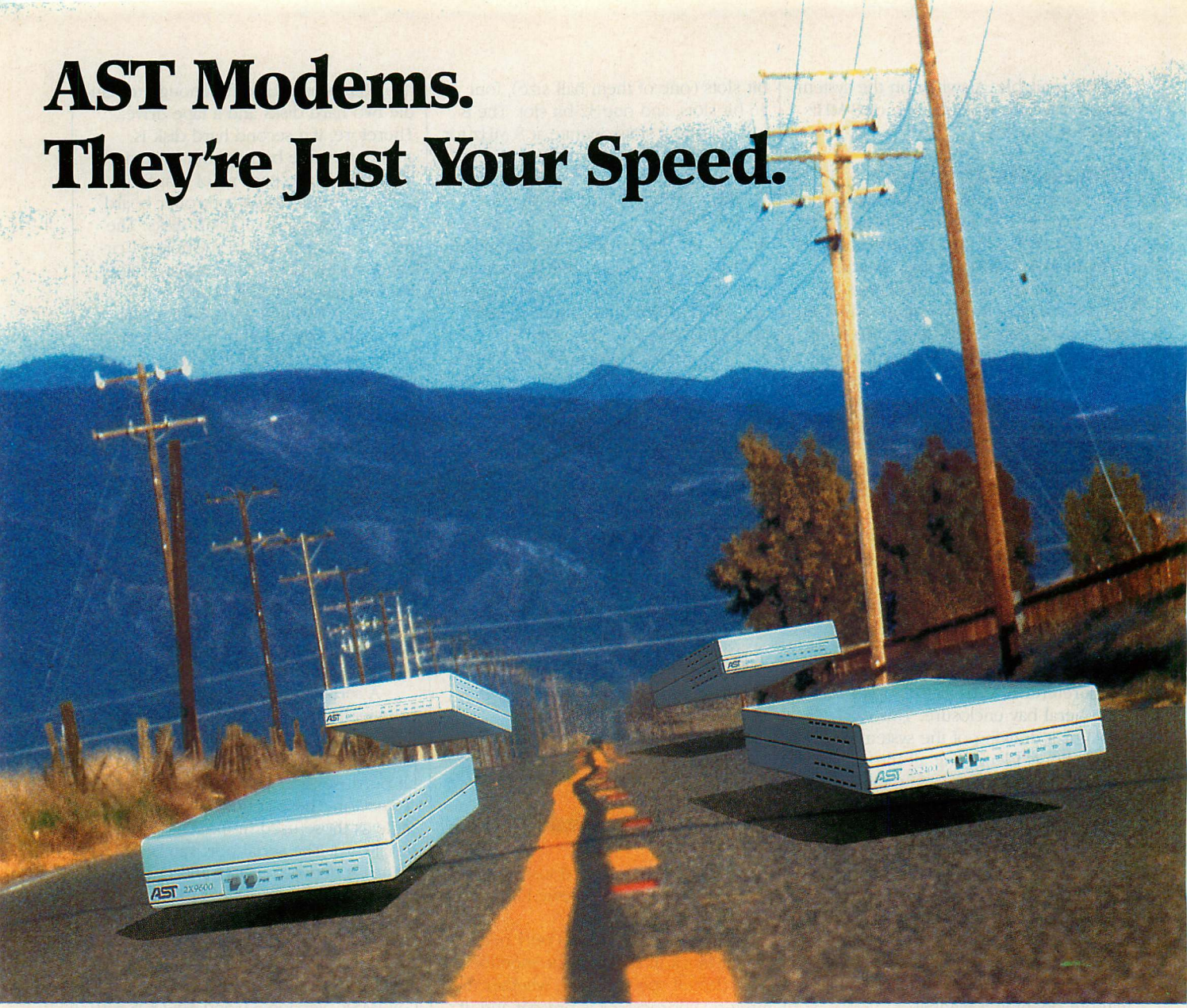
STANDARD SOFTWARE

GW-BASIC version 3.0 and MS-DOS 3.10 are available at extra cost with the Deskpro 386. Included with this package is a DOS 3.2 support disk that contains a replacement set of disk utilities that works with IBM's PC-DOS 3.2 and is aware of the Deskpro 386's 40MB, 70MB, and 130MB hard disks.

Compaq's DOS 3.1 disk contains all the items that are contained in IBM's DOS 3.1, plus some extras. One of the added files is DISKINIT, a utility that simplifies setting up a hard disk. It performs the FDISK, ENHDISK, and FORMAT functions, creates CONFIG.SYS and AUTOEXEC.BAT files, and copies both DOS diskettes into a subdirectory called /DOS. These operations are similar to the ones that are performed by the SELECT command in DOS 3.2.

Other extra programs include SETCLOCK, which allows users to reset the date and time without running the set-up program; CMPQADAP, a utility for modifying the keyboard driver for

AST Modems. They're Just Your Speed.



AST introduces a new family of standalone and internal PC modems designed for the way you work.

Whether you're shopping for a state-of-the-art, high-speed 19.2K bps modem with error correction; a low-cost 1200 bps, Hayes™ compatible modem loaded with features…or, anything in between…AST has just your speed.

AST's new modem family includes 1200, 2400, 4800 and 19,200 bps modems in both standalone (synchronous and asynchronous) and internal PC (asynchronous) models.

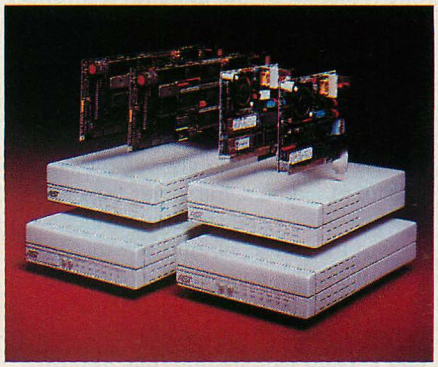
Fast and and Reliable.

The AST-2X9600™ standalone and AST-2X9600™ internal PC modems save thousands of dollars per year in telephone-line charges when used for intensive, high-volume file transfer applications. Fast, 19,200 bps speed is achieved using the industry standard MNP™ data compression. While data integrity is assured with MNP error correction.

For less intense applications, the AST-2X2400™ and AST-2X2400B™ provide data compression and error correction at speeds up to 4800 bps.

Low-Cost, Hayes-Compatible.

The AST-1200™ and AST-1200B™ provide reliable, 1200 bps dial-line communications for a variety of applications, including popular interactive and bulletin-board services.



Faster, yet price competitive with many 1200 bps modems, are AST's two 2400 bps modems—AST-2400™ and AST-2400B.™ They're the right solutions for a mix of interactive applications and low-volume file transfers.

Ask for AST modems by name. Visit your local, authorized AST dealer and compare AST modem prices and features. There's an AST modem just your speed.

Send today for AST's helpful Modem Selection Guide. It'll help you select the modem that's right for your particular application. For your free copy of this guide send the attached coupon to AST Research, Inc., 2121 Alton Ave., Irvine, CA 92714-4992, Attn. M C or call **(714) 863-1480**.

AST® RESEARCH INC.

Yes! Send me information on AST Modems today!

Name_____

Title_____

Company_____

Address_____

City/State/Zip_____

Telephone_____

01PCTE086A01MD

PCTJ 3/87

Send to: AST Research, Inc., 2121 Alton Ave. Irvine, CA 92714-4992, Attn: M C

AST markets products worldwide—in Europe call: 44 1 568 4350; in the Far East call: 852 0499 9113; in Canada call: 416 826 7514. AST registered and AST-1200, AST 1200B, AST-2400, AST-2400B, AST-2X2400, AST2X2400B. AST-2X9600 and AST-2X9600B trademarks of AST Research, Inc. MNP trademark of Microcom, Inc. Hayes trademark of Hayes Computer Products. Copyright © 1986 AST Research, Inc. All rights reserved.

international character sets; KEBDA, KEYBNO, KEYBSU, and KEYBSV, which are four additional keyboard drivers; GRAFTABL, which enables the color/graphics adapter when operating in graphics mode to display characters whose ASCII values are greater than 127; TAPE, a cartridge tape utility that formats a tape, backs up files to tape, restores files, lists a tape directory, and erases tapes; and SETUP, which stores configuration information in the memory of the realtime clock chip.

Software supplied with the Deskpro 386 includes ENHDISK.SYS, a driver that assigns a separate drive letter to each partition of a hard disk. This is necessary for using all the storage on the Deskpro 386's large hard disks, because DOS limits partitions to 32MB. Two CEMM programs (CEMM.EXE and CEMM.COM) are supplied, as well as INST386, which generates or modifies the CONFIG.SYS file in order to identify CEMM.EXE and VDISK.SYS as device drivers and indicate the amount of memory to be used for expanded memory and virtual disks.

Finally, Microsoft provides a diagnostic program called TEST that is the equivalent of IBM's standard diagnostics. Curiously, this program contains a test for the Microsoft InPort mouse. Compaq was rumored to be including an InPort connector on its Enhanced Graphics Board, but it is not contained on the released version. Compaq may have forgotten to remove the diagnostic test when it decided not to include the InPort connector.

Compaq continues to score high marks with its technical documentation. Included with every Deskpro 386 is a useful *Operations Guide*. Users also can purchase the excellent two-volume *Technical Reference Guide*.

The *Technical Reference Guide* is one of the finest works of its kind and an essential purchase for anyone who is curious about how the Deskpro 386 really works. It is clearly written and provides enough background material so that even novices can understand the complex topics it covers. The explanation of the 80386's B1 stepping is especially good. Unfortunately, much of the information that IBM places in its *Guide to Operations* manuals is available from Compaq only in this separately priced reference. The explanation of switch settings for the multipurpose disk controller board is one example.

Compaq produces a single set of DOS documentation for all Compaq computers, with special supplements for the Deskpro 386. While this may be

TABLE 2: *Disk Drives Supported*

DRIVE TYPE	NO. OF CYLINDERS	NO. OF HEADS	CAPACITY (MB)	LANDING ZONE CYLINDER	PRECOMPENSATION CYLINDER	SECTORS/TRACK
1	306	4	10.65	305	128	17
2	615	4	21.41	638	128	17
3	615	6	32.12	615	128	17
4	1,024	8	71.30	1,023	512	17
5	940	6	49.09	939	512	17
6	697	5	30.33	696	128	17
7	462	8	32.17	511	256	17
8	925	5	40.26	924	128	17
9	900	15	117.50	899	−1[a]	17
10	980	5	42.65	980	−1	17
11	823	10	71.63	822	−1	17
12	925	9	72.46	924	128	17
13	612	8	42.61	611	256	17
14	754	11	72.19	753	−1	17
15	0	0	0	0	0	0
16	612	4	21.31	612	0	17
17	980	5	42.65	980	128	17
18	966	6	50.45	966	128	17
19	1,023	8	71.23	1,023	−1	17
20	733	5	31.90	732	256	17
21	733	7	44.66	732	256	17
22	768	6	40.11	768	−1	17
23	771	6	20.16	771	−1	17
24	966	14	117.71	966	−1	17
25	966	16	134.53	966	−1	17
26	1,023	14	124.66	1,023	−1	17
27	966	10	84.08	966	−1	17
28	771	3	20.11	771	−1	17
29	578	4	20.09	578	−1	17
30	615	4	31.49	615	128	25
31	615	8	62.98	615	128	25
32	966	3	50.45	966	−1	34
33	966	5	84.08	966	−1	34
34	966	7	117.71	966	−1	34
35	966	8	134.53	966	−1	34
36	966	9	151.35	966	−1	34
37	966	5	84.08	966	−1	34
38	1,023	9	155.56	1,023	−1	33
39	1,023	11	190.13	1,023	−1	33
40	1,023	13	224.70	1,023	−1	33
41	1,023	15	259.27	1,023	−1	33
42	1,023	16	284.93	1,023	−1	34
43	756	4	40.26	756	−1	26
44	756	2	20.13	756	−1	26
45	756	4	40.89	768	−1	26
46	768	2	20.45	768	−1	26
47	966	5	61.82	966	128	25

[a]The −1 values mean that no write precompensation is used for the drive.

The Deskpro 386 BIOS supports 47 drive types. These include all the drive types (with the same type numbers) supported by the IBM AT, plus many more. This allows a high degree of flexibility in the selection of disk drives.

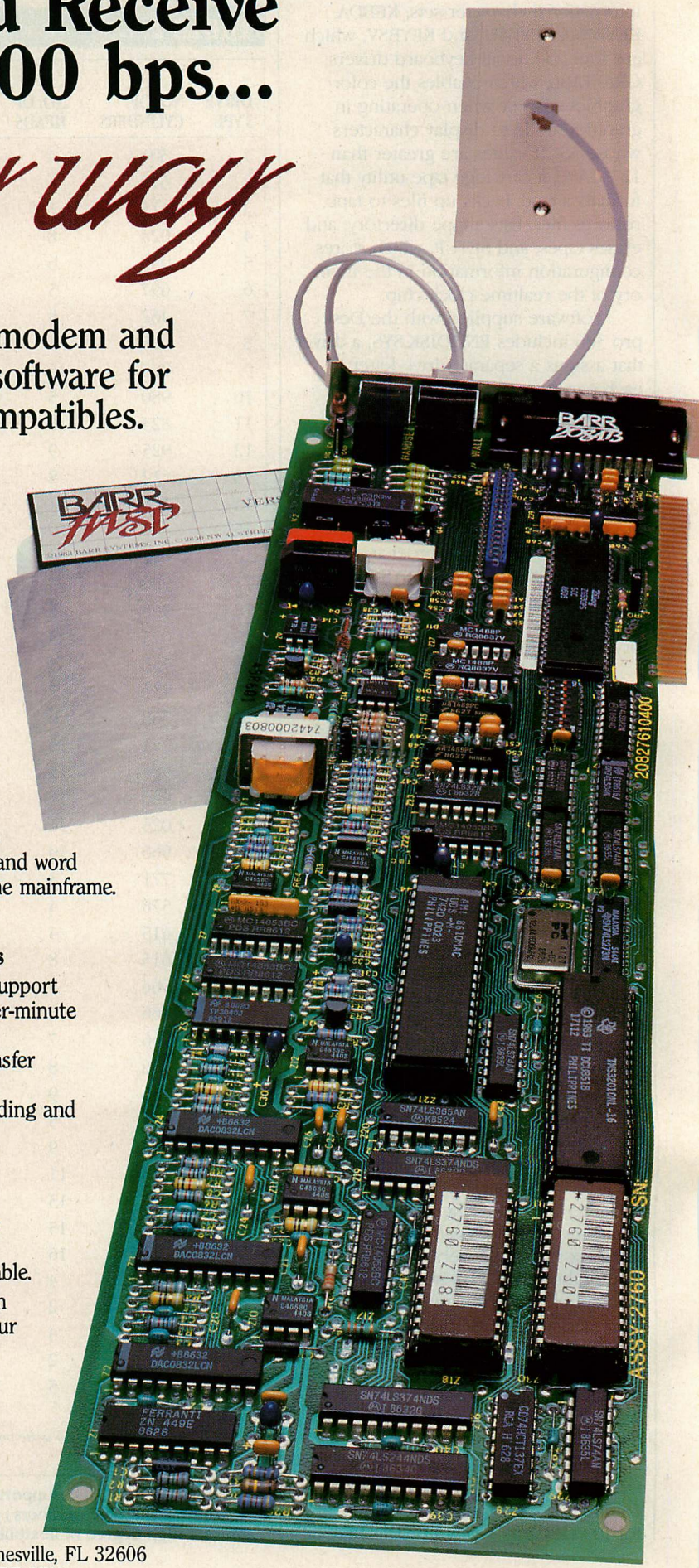
Send Data and Receive Reports at 4800 bps...

the easy way

BARR/208AB synchronous modem and BARR/HASP multi-tasking software for the IBM PC, XT, AT and compatibles.

Just insert the 208AB modem card, install the menu-driven software, and instantly the PC becomes an RJE workstation. Plug the cable into the telephone outlet, send a command, and receive an immediate response from the mainframe—invoices, production reports, laboratory analyses, checks...all done for you automatically. The BARR/HASP software bypasses the high-overhead TSO and CMS interactive facilities, and instead uses the mainstream communications facilities of the host —MVS/JES, VM/RSCS, CDC/NOS/RBF—saving time and money.

What's more, BARR/208AB and BARR/HASP were designed for beginners and experts alike with all features quickly accessed through simple menus. And the PC is always free for data entry and word processing, even while it's communicating with the mainframe.



BARR/208AB
Modem Features

- Auto-dial
- Auto-answer
- Advanced digital signal processing
- Automatic night dialing
- Self-test diagnostics

BARR/HASP
Software Features

- Multiple printer support up to 600 lines-per-minute
- LAN gateway
- PC to PC file transfer
- Concurrent DOS
- Simultaneous sending and receiving files

- Bell 208A/B compatible modem
- Emulates IBM 3777-2 and HASP on IBM 360/20

The complete communications package includes BARR/208AB modem, BARR/HASP software, manuals, and cable. $2290. Trial evaluation available.

Call BARR SYSTEMS today for more information about how this great combination can help in your business. 800-BARRSYS

BARR

Breaking the limits of micro-to-mainframe communications

BARR SYSTEMS, INC. ☐ 2830 NW 41 Street ☐ Gainesville, FL 32606
904-371-3050 ☐ 800-BARRSYS ☐ Telex 6715114 BARR UW

CIRCLE NO. 147 ON READER SERVICE CARD

convenient for Compaq, it creates a great deal of confusion for the user. The DOS 3.1 documentation and diskettes form a generic package that is available with all Compaq computers. Two other diskettes, the User Prog and Supplemental User Prog diskettes, contain additional utilities as well as updated versions of certain programs and versions designed especially for the Deskpro 386. For example, the User Prog diskette contains a new version of the TAPE program that must be used instead of the version on the DOS diskettes. The Supplemental User Prog diskette contains versions of VDISK and the MODE command designed for the 80386 processor. Nowhere does the DOS manual mention the existence of these updated programs. Nor does the DOS manual describe the CEMM program. The Deskpro 386 *Operations Guide* mentions the new VDISK, CEMM, and MODE versions, but not the new TAPE command. With multiple copies of programs and utilities, the user can easily install the wrong versions. The manuals do nothing to explain the situation or ease the problem.

Instead of the usual 90-day warranty, Compaq provides a one-year warranty for the Deskpro 386, including the Enhanced Graphics Board and Compaq Color Monitor. If users experience problems during that time, they can return the machine to their dealer for repairs or replacement.

TEST RUN

The Deskpro 386 was subjected to the same tests used throughout this series on AT compatibles, although they were modified to run on 80386-based computers. First, commonly used software and hardware products were installed to check for compatibility. Then the *PC Tech Journal* AT Evaluation Suite of compatibility and performance tests was run and the results were compared with those of the 8-MHz IBM AT. (For details on how these tests were modified, see "Updating the Evaluation Suite," Ted Forgeron, Paul Pierce, and Steven Armbrust, this issue, p. 70).

All of the hardware products tested with the Deskpro 386 worked without problem, including an 80287 numeric coprocessor, an Intel Above Board with 4MB of memory, Microsoft serial and bus mice, a Hayes Smartmodem 1200, and the Compaq Enhanced Graphics Board. One item worth special mention is that expanded memory on the Above Board (or on any expanded memory board) cannot be used when Compaq's CEMM driver is installed. The reason for this is simply because only one expanded memory driver can be active at any one time in one computer.

Software installed in the Deskpro 386 included Microsoft Windows 1.01 and Microsoft Word 3.1 to test graphics capabilities and the mouse; Borland's SuperKey 1.11A, SideKick 1.56A, and Turbo Lightning 1.101A to test memory-resident programs; Living Videotext's Ready! 1.00C and Intel's QUIKMEM2 RAM disk to test expanded memory; both the IBM and Compaq versions of VDISK to check the computer's ability to switch in and out of protected mode; Fastback from Fifth Generation Systems to test direct memory access; and the IBM Advanced Diagnostics programs to perform a general check-up.

The Technical Reference Guide is an essential purchase for anyone who is curious about how the Deskpro 386 really works.

Almost all the programs worked without error. Only SuperKey failed completely by locking the enhanced keyboard, but a new version (1.15) released specifically to solve this problem worked as advertised on the Deskpro 386. Fastback performed without errors, but the restore program required the processor speed to be set to 8 MHz. The only other program that showed a discrepancy was the IBM Advanced Diagnostics program, which recognized only 640KB of the Deskpro 386's first 1MB of RAM. The remaining 360KB is mapped to addresses that are just below 16MB, where the Diagnostics program could not find it.

The expanded memory programs (Ready! and QUIKMEM2) were tested by running them out of CEMM memory; both ran without problems.

The IBM VDISK program also worked as intended. This is significant because it implies that even though the 80386 can easily switch between protected and real modes, the Deskpro 386 also supports the more complicated method required for the 80286 (which involves resetting the processor). The Deskpro 386's VDISK program is written especially for the 80386 processor and does not need to reset the processor to go back to real mode.

One other program tested on the Deskpro 386 did not run correctly. Gold Hill Computers' Golden Common LISP (GCLISP) large memory version 2.0, a program that switches in and out of protected mode to execute code out of extended memory, caused the computer to perform a warm reboot. Gold Hill attributes the problem to the aggressive, nonstandard way that GCLISP uses 80286 protected mode. Gold Hill plans to market a Deskpro 386-compatible version of GCLISP in 1987.

The *PC Tech Journal* compatibility and performance tests consist of five programs: ATBIOS checks the BIOS and BIOS data areas; ATKEY checks for keyboard compatibility; ATFLOAT measures floating-point operations with the numeric coprocessor installed; ATDISK measures hard-disk performance; ATPERF measures processor and numeric coprocessor clock rates, as well as memory access times. Table 3 lists the results of these tests, and figure 5 presents bar-graph comparisons between the 8-MHz AT and Deskpro 386 for RAM instruction fetch, RAM read and write times, ROM read times, and EMM read and write times.

ATBIOS showed that the Deskpro 386 uses the BIOS data area in the same way as the AT. It listed a Compaq copyright and a date of 8/19/86.

ATKEY verified the keyboard compatibility. The original AT keyboard also worked with the Deskpro 386.

ATFLOAT showed that the Deskpro 386 can process floating-point operations 1.7 times as fast as an 8-MHz AT. Given that the processor runs at 16 MHz and the numeric coprocessor at 8 MHz (compared to 8 and 5.33 on the AT), this value seems appropriate.

ATDISK proved that the 40MB drive in the Deskpro 386 is a high-performance model consistent with the requirements of an 80386-based computer. It showed considerably better access times than the IBM drive, and it is formatted and works effectively with an interleave of 2, causing a higher effective transfer rate than the IBM drive (which uses an interleave of 3).

The most interesting results were obtained from ATPERF, which demonstrated that the Deskpro 386 is an extremely fast computer, with instruction fetches, as well as memory reads and writes, almost three times as fast as an 8-MHz AT. (Because the system normally runs at 8 MHz when the diskette drive is accessed, ATPERF must be run from the hard disk or SPEED must be set to HIGH using the MODE command to achieve these results.)

The RAM read and write times and the corresponding wait states show the values expected from a high-performance, 32-bit processor. ATPERF shows that reading from and writing to RAM within a 2KB page indeed occurs at zero wait states.

The modified version of ATPERF uses PUSHA instructions to measure RAM writes. These are fast instructions that do not result in idle bus cycles. Each PUSHA instruction involves eight PUSH instructions (pushing all of the registers), and therefore it writes to eight consecutive memory locations that are almost always within the same 2KB page. Memory is accessed outside the page when the processor fetches a new instruction. However, each PUSHA is a byte instruction, so each instruction fetch during this test fetches four separate PUSHA instructions (the 80386 accesses four bytes at a time). As a result, each instruction fetch results in 32 memory accesses (eight pushes from each of four PUSHA instructions). With this mechanism, the effect of accessing memory outside of a page during instruction fetch is barely noticeable and is not enough to affect the number of wait states measured.

The ROM read times are also worth discussing. ATPERF checks the ROM read time by using MOVS instructions to copy the system BIOS to RAM. However, in the Deskpro 386 the system BIOS reads are actually performed from RAM instead of ROM. Therefore, the ROM read times are, in fact, equal to the RAM read times as indicated in table 3, with the actual performance dependent upon the extent to which successive memory accesses occur within the same 2KB page.

The ROM read test is a general-purpose test that is not customized for the Deskpro 386. It subtracts the average RAM write time that was discussed previously from the average time required to copy ROM to RAM in order to determine the average ROM read time. In the case of the Deskpro 386, the average RAM write time is measured at zero wait states, whereas the RAM writes in the ROM copy operation occur at two wait states (because the MOVS source and destination addresses are not in the same 2KB page.) Therefore, subtracting the zero-wait-state average write time yields an inaccurate value for average ROM read time.

The value that is displayed (0.39 microseconds) is quite useful, however, for determining the average time needed for RAM read/write accesses that are not contained within the same 2KB

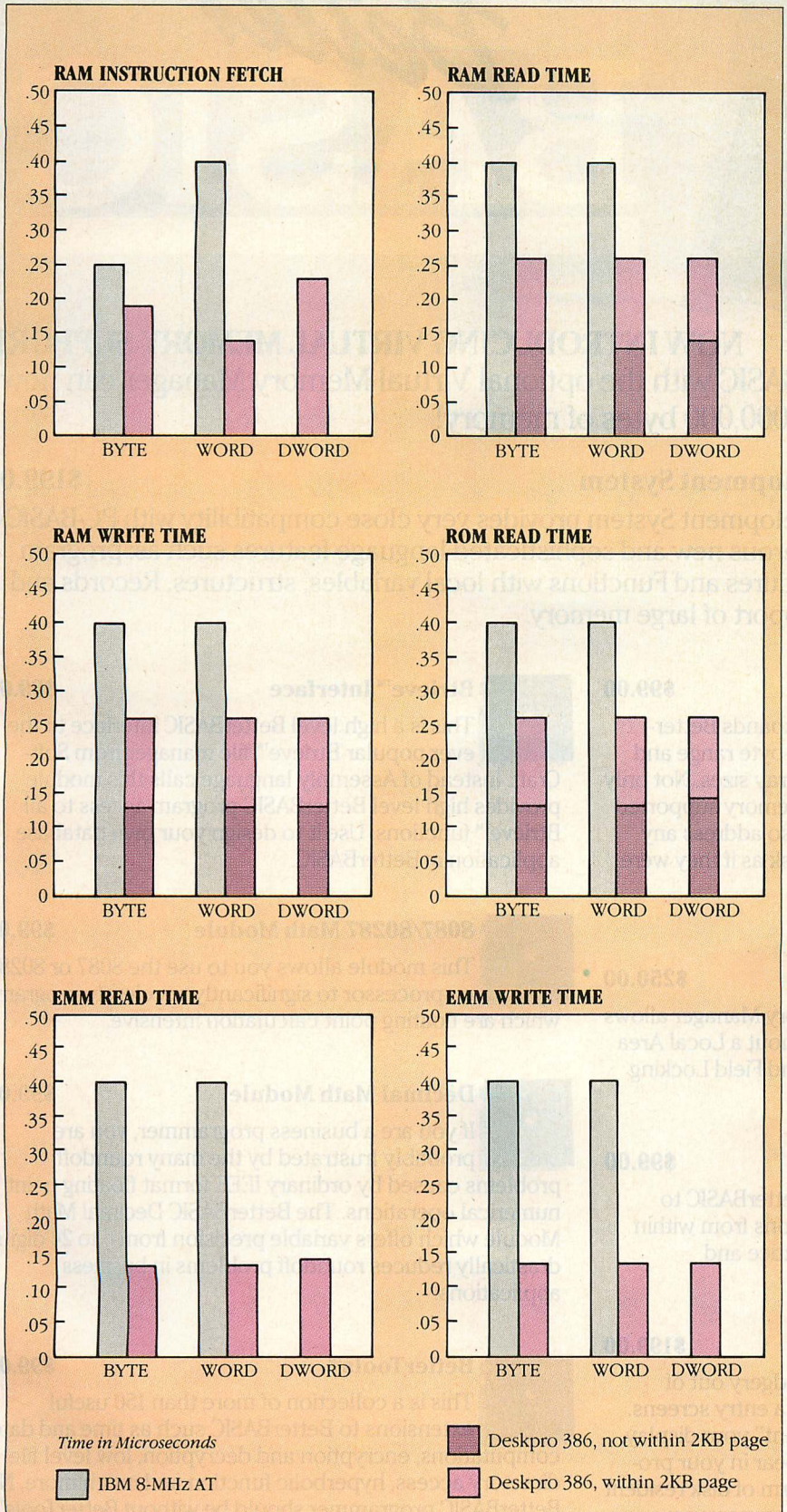
TABLE 3: *Compatibility and Performance Tests*

	8-MHz AT, 30MB DISK[a]	DESKPRO 386, 40MB DISK
ATBIOS		
ROM BIOS date	11/15/85	08/19/86
ATPERF		
Average RAM instruction fetch (μs)		
BYTE	.250	.19 (130)[b]
WORD	.403	.14 (280)
DWORD	N/A	.23
Average RAM read time (μs)[c]		
BYTE	.401	.13/.26 (298/154)
WORD	.401	.13/.26 (298/154)
DWORD	N/A	.14/.26
Average RAM write time (μs)[c]		
BYTE	.401	.13/.26 (307/154)
WORD	.401	.13/.26 (307/154)
DWORD	N/A	.13/.26
Average ROM read time (μs)		
BYTE	.401	Same as RAM read
WORD	.401	Same as RAM read
DWORD	N/A	Same as RAM read
Average CGA video write time (μs)		
BYTE	1.208	1.21 (100)
WORD	2.415	2.42 (100)
DWORD	N/A	4.83
Average EMM read time (μs)		
BYTE	.402	.13 (301)[d]
WORD	.402	.13 (301)[d]
DWORD	N/A	.14
Average EMM write time (μs)		
BYTE	.402	.13 (306)
WORD	.402	.13 (306)
DWORD	N/A	.13
CPU clock rate (MHz)	8.0	16.0 (200)
Numeric coprocessor clock rate (MHz)	5.3	8.0 (150)
Refresh overhead (%)	7.1	15
RAM read/write wait states	1/1	0/0
ROM read wait states	1	2
Video write wait states (CGA)	8	17
EMM read/write wait states	1/1	0/1[d]
ATFLOAT		
Performance as percentage relative to AT	100	107
ATDISK		
Sectors/track	17	17
Heads	5	5
Cylinders	731	978
Total space (million bytes)	31.81	42.56
Track-track seek time (ms)	6.0	4.1
Average seek time (ms)	37.1	27.6
Effective transfer rate (KB/sec)	170.1	254.5
DOS file I/O (sec)	7.3	7.4
Interleave	3	2

[a]The figures for the IBM AT are the average results from several machines, whereas the results from the Compaq Deskpro 386 are taken only from the review sample model.
[b]Figures shown in parentheses represent the relative performance expressed as a percentage compared to PC Tech Journal*'s baseline machine, the 8-MHz, 30MB AT.*
[c]For the Deskpro 386, first number is for memory access within the same 2KB page; the second is not within the same 2KB page.
[d]EMM measurements were taken using the Deskpro 386's CEMM driver and built-in memory.

RAM read/write times are from 1.5 to 3 times faster that for the AT, depending on whether or not successive accesses are written to the same 2KB page.

FIGURE 5: *Performance Comparison*



The Deskpro 386 can access a 32-bit double word faster than the AT can perform even one of the two 16-bit word accesses that it uses to perform the same task.

page. This value is determined by adding the previously subtracted average RAM write time to the average ROM read time displayed by ATPERF and then dividing that total by two (because the RAM reads and writes take the same amount of time). The resultant 0.26 microseconds per double word is the expected average time for a two-wait-state RAM read/write.

FASTEST ON MARKET

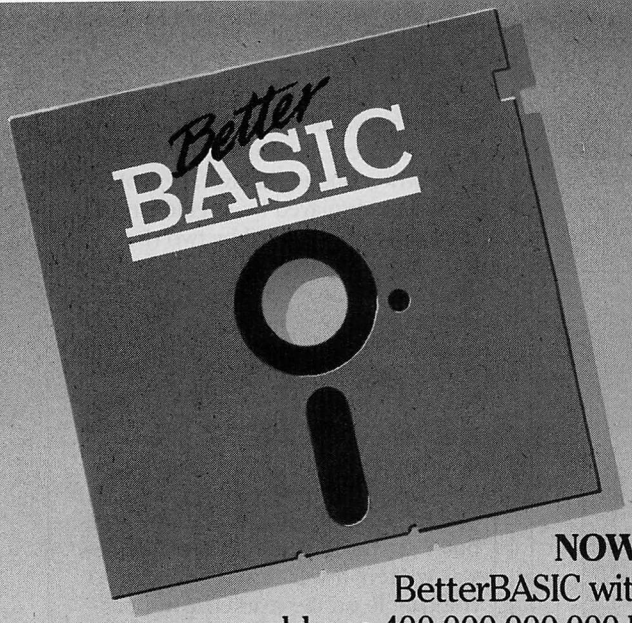
For anyone who demands high performance and AT compatibility, the Compaq Deskpro 386 is definitely the computer of choice. At the time this review was written, it was easily the fastest AT compatible on the market, as well as one of the most compatible. The addition of Compaq's special touches, such as processor speed control, CEMM, and excellent documentation, make this computer even more desirable. Perhaps its best and brightest feature is its 80386, which offers a bridge between today's real-mode applications and tomorrow's more powerful multitasking and protected mode programs.

Only a few areas need improvement in the Deskpro 386. The absence of a socket for the 80387 numeric coprocessor will keep this machine's floating-point performance less than that provided by other new 80386-based machines that do support the 80387. Even other computers that contain 10-MHz 80287s will be able to edge the Deskpro 386's 4- and 8-MHz 80287 in floating-point performance. Other areas of concern are the tape backup drive, because one of the units examined did not work properly, and the Compaq Color Monitor, because it does not match the quality of some of the other enhanced color monitors that are currently on the market.

These are problems, however, that can be remedied, and they are far outweighed by the outstanding performance and quality of the Deskpro 386. Even IBM will have a hard time topping this machine.

Compaq Computer Corp.
2033 FM 149
Houston, TX 77070
713/370-0670
Deskpro 386
CIRCLE 342 ON READER SERVICE CARD

Steven Armbrust is a freelance technical writer, and Ted Forgeron is software project manager for Intel Scientific Computers. Together, they are the authors of the Programmer's Reference Manual for IBM Personal Computers *(Dow-Jones Irwin, 1986).*



Better BASIC

NOW INTRODUCING VIRTUAL MEMORY SUPPORT

BetterBASIC with the optional Virtual Memory Manager can now address 400,000,000,000 bytes of memory!

BetterBASIC Application Development System **$199.00**

The BetterBASIC Application Development System provides very close compatibility with PC-BASICA and GW-BASIC, yet provides numerous new and sophisticated language features such as: program Block Structures, recursive Procedures and Functions with local variables, structures, Records and Pointers and last but not least support of large memory.

Virtual Memory Manager **$99.00**

The Virtual Memory Manager expands BetterBASIC's data space into the giga-byte range and finally breaks the 640k byte barrier for array sizes. Not only can you directly address all expanded memory supported by LIM/EMS memory boards, you can also address any RAM Disk, Hard Disk or even a Floppy Disk as if they were ordinary RAM.

Virtual Memory Manager-Network Version **$250.00**

This version of the Virtual Memory Manager allows Virtual Memory to be distributed throughout a Local Area Network. It also provides File, Records and Field Locking to control access to shared data.

C-Link **$99.00**

This software package allows BetterBASIC to access C-language library functions from within BetterBASIC. Currently supported are Lattice and Microsoft C.

Screen Design System **$199.00**

This package truly takes the drudgery out of creating display screens and data entry screens. An interactive Screen Editor lets you "paint" your display screens exactly as you want them to appear in your program. The completed screens take the form of disk resident images. A run time library module provides many new BetterBASIC procedures and functions for interacting with the display screens to simplify the use of pop-up menus and data entry screens.

Btrieve™ Interface **$99.00**

This is a high level BetterBASIC interface to the ever popular Btrieve™ file manager from SoftCraft. Instead of Assembly language calls this module provides high level BetterBASIC program access to all Btrieve™ functions. Use it to design your own database application in BetterBASIC.

8087/80287 Math Module **$99.00**

This module allows you to use the 8087 or 80287 co-processor to significantly accelerate programs which are floating point calculation intensive.

Decimal Math Module **$99.00**

If you are a business programmer, you are probably frustrated by the many roundoff problems caused by ordinary IEEE format floating point numerical operations. The BetterBASIC Decimal Math Module which offers variable precision from 6 to 24 digits, drastically reduces roundoff problems in business applications.

BetterTools™ **$99.00**

This is a collection of more than 150 useful extensions to BetterBASIC such as time and date computations, encryption and decryption, low level file directory access, hyperbolic function and much more. No BetterBASIC programmer should be without BetterTools™.

CIRCLE NO. 195 ON READER SERVICE CARD

SPECIFICATIONS

BetterBASIC is GW-BASIC and PC-BASICA compatible; runs on IBM PC and compatibles.

HARDWARE REQUIREMENTS

CPU: IBM PC, IBM PC XT AT, COMPAQ, IBM PC Compatibles
Memory: 256KB min up to 640KB
Display: Monochrome or Color
Disk Drive: One 5¼" floppy, single or double sided
Operating Systems: MS-DOS 2.0, 2.1, 3.0, 3.1

DATA TYPES:

Numeric Data:
BYTE, range: 0 to +255
INTEGER, range: − 32768 to + 32767
REAL, range: Single Precision $8.43x10^{-37}$ to $3.37x10^{38}$
Double precision $4.19x10^{-307}$ to $1.67x10^{308}$
Binary Math, Single/Double/Mixed Precision
Mixed mode numeric expressions will always be REAL.

String data:
Variable from 0 to 32767 characters in size.

Record Variables:
Allows grouping of dissimilar data types into a single logical variable. Elements of a RECORD are addressed as FIELDS and can be of any type, including ARRAY, RECORD and POINTER.

Array Variables:
N-dimensional arrays of any type, including ARRAY, RECORD and POINTER. Dynamic arrays like PC-BASICA

Pointer Variables:
Allows indirect reference to any data type. Can be used with RECORD variable to create linked lists, or to create relational data structures.

In addition supports PC-BASICA record types.

BetterBASIC BENCHMARK COMPARISON

in milliseconds

	Better BASIC			IBM INTERPRETIVE		IBM COMPILED	
	SP*	DP*	8087 DP	SP	DP	SP	DP
REAL FOR/NEXT	1.3	1.4	0.55	0.93	0.93	0.7	0.7
ASSIGNMENT	1.0	1.0	0.93	1.5	1.5	0.1	0.1
ADD	0.77	1.1	0.44	1.6	2.3	0.4	0.4
MULTIPLY	0.88	1.8	0.49	1.9	3.0	0.5	0.8
DIVISION	1.0	3.0	0.49	2.8	19.7	0.6	1.1
LOGARITHM	5.7	15.6	0.55	7.5	64.0	4.0	11.9
EXPONENTIAL	7.4	27.0	0.66	6.5	43.0	3.6	10.8
SINE	4.7	17.0	0.82	17.6	35.0	3.2	12.4
COSINE	4.5	17.0	0.77	25.0	41.0	3.5	12.7
TANGENT	7.2	18.0	0.66	44.0	94.0	6.9	26.0
X^Y	13.8	44.5	1.1	15.2	115.0	7.7	24.0
SQR (SQUARE ROOT)	1.4	6.5	0.33	7.2	95.0	1.1	3.5

*SP = Single Precision
DP = Double Precision

ADDITIONAL BetterBASIC STATEMENTS

ANY ARG
APPEND
ASH
ASSIGN
AUTODEF
BIN$
BREAK
BREAK OFF
BYE
BYT
BYTE
BYTE ARG
BYTE ARRAY
BYTE ARRAY ARG
BYTE ARRAY PTR
BYTE ARRAY STRUC
BYTE PTR
CHANGE
CHAR$
CHECK
CLD
CLW
CODE
COLOR BORDER
COMMAND$
COMPRESS
CONSTANT
DEFINE WINDOW
DEL$
DIR$
DISABLE
DO
DO IF
DO UNTIL
DO X TIMES
DRIVE$
DYNAMIC
END DO
END FUNCTION
END PROCEDURE
ENDPROC
ERRORMODE
EXIT
EXIT GOSUB
EXIT X LEVELS
EXTERNAL
FRAME
FRAME WINDOW
FREEDISK
GOTO END
HEADER
INPUT FROM
INS$
INTEGER
INTEGER ARG
INTEGER ARRAY
INTEGER ARRAY ARG
INTEGER ARRAY PTR
INTEGER ARRAY STRUC
INTEGER FUNCTION
INTEGER PTR
INTERRUPT
INTERRUPT CLEAR
INTERRUPT ON/OFF
INTERRUPT PROC
INTERRUPT RESTORE
INTERRUPT SAVE
INTR
KEY=
KEYWORD ARG
KEYWORD SET
LIST ALL
MAIN
MAKE MODULE
MAKE PROGRAM
MAX
MAX$
MEM
MIN
MIN$
MODULES=
OFFSET
ON INTERRUPT
PRECISION=
PRINT TO
PRINT TO USING
PROCEDURE
PROCS=
PUBLIC
READ RECORD
READCHR
READCHR FROM
READLINE
READLINE FROM
READ RECORD
REAL
REAL ARG
REAL ARRAY
REAL ARRAY ARG
REAL ARRAY STRUC
REAL FUNCTION
REAL PTR
RENAME
REPEAT
RESTART
RESTORE PAR
RESTORE SCREEN
RESULT=
RETRY
ROT
SAVE MODULE
SAVE PAR
SAVE SCREEN
SCOPE=
SCRATCH
SEG
SELECT
SET
SET CURSOR
SH
SHELL
SIZE
SIZE$=
SPAN
STACK=
STATUS=
STATUSLINE
STRING
STRING ARG
STRING ARRAY
STRING ARRAY ARG
STRING ARRAY PTR
STRING ARRAY STRUC
STRING FUNCTION
STRING PTR
STRUCTURE
SYSCALL
SYSCODE
SYSFLAGS
TYPE
UPPER$
WHILE...DO
WINDOW
WOR
WRITE RECORD
WRITE TO
XMEM
XMEM=
XREF

Microsoft Statements Not Supported

DEF USR
MERGE
MOTOR
ON PEN
ON STRIG
PEN
STICK
STRIG
USR

Call our Toll Free Order Line

1-800-225-5800



Summit Software Technology, Inc.™
106 Access Road
Norwood, MA 02062
(617) 769-7966

BetterBASIC is also available from TANDY/RADIO SHACK Computer Centers. Ask your dealer for Express Order Software #90-0315 for BetterBASIC, and #90-0303 for Runtime System.

BetterBASIC is a registered trademark of Summit Software Technology, Inc. IBM PC, XT, AT, are registered trademarks of International Business Machines Corp. Tandy is a registered trademark of Tandy Corp. Lotus™ and 1-2-3™ are registered trademarks of the Lotus Development Corp. Intel © Intel Corp. HALO © Media Cybernetics, Inc. GSS © Graphic Software Systems, Inc. Lattice *C* is a registered trademark of Lattice, Inc. Btrieve is a registered trademark of SoftCraft Inc.



Updating the Evaluation Suite

As the state of the art progresses, so must our testing procedures. PC Tech Journal's compatibility and performance metrics have been expanded to include other processors.

TED FORGERON, PAUL PIERCE, and STEVEN ARMBRUST

For several months, *PC Tech Journal* has been reviewing 80286-based computers, examining whether or not they are compatible with the IBM PC/AT and how they perform compared to the 8-MHz AT, as well as the characteristics that make these computers unique. An integral part of these reviews has been a group of programs referred to as the AT Evaluation Suite. These programs were introduced in the article "Out from the Shadow of IBM. . .," (Steven Armbrust, Ted Forgeron, and Paul Pierce, August 1986, p. 52), which described the tests and listed their source code.

Now that Compaq has introduced an AT compatible based on Intel's new high-performance 80386, the tests had to be revised in order to measure the performance of 386-based machines.

REVISED TESTS

The compatibility tests consist of five separate programs. ATBIOS examines the computer's BIOS. ATKEY tests keyboard compatibility and programmability. ATPERF measures processor and memory performance. ATFLOAT measures the performance of floating-point operations. ATDISK measures hard-disk performance.

Because these programs were originally designed to test AT-compatible computers, they assumed that each computer contained an 80286 micro-

PHOTOGRAPH • MARC DAVID COHEN

processor. In particular, ATPERF calculated many of its results by assuming the number of clock cycles required for particular instructions. Because this number can vary from processor to processor, ATPERF gave valid results only when the computer on which it ran contained an 80286. Revised listings for ATPERF are included in this article. Small changes also have been made to the other programs to correct minor problems, but only the affected sections of code are given here. Complete listings for all the programs are available on PCTECHline.

ATBIOS. ATBIOS has been updated to version 1.01 by correcting the problems that were previously reported in the Letters column ("The Shadow of a Doubt," December 1986, p. 16). Three changes have been made.

First, when control characters appeared in the range of memory that ATBIOS checked for copyright messages, the program sometimes printed random characters to the screen instead of the copyright statement. Version 1.01 filters out control characters by displaying them as periods, much as DEBUG handles nonprinting characters. The fix involves removing the following lines:

```
write ('Copyright Statement is ');
window (40,6,80,7);
gotoxy (1,1);
write (copyright);
window (1,1,80,25);
gotoxy (1,8);
```

and replacing them with these lines:

```
write ('Copyright Statement is ');
write ('                        ');
FOR i := 1 TO 80 DO
  BEGIN
    IF (copyright [i2H < ' ')
    OR (copyright2Gi2H > '~') THEN
      write ('.')
    ELSE
      write (copyright[i]);
    IF i = 40 THEN
      BEGIN
        writeln;
        write ('                        ');
        write ('                        ');
      END;
    END;
  writeln;
```

The second problem involved ATBIOS's inability to recognize the IBM game adapter, which the AT does not support. Because the AT never sets the game adapter bit in the equipment flag, ATBIOS was fooled into thinking the game adapter was not present, even when it was installed. The new version now checks that bit only for non-AT machines. The change involves removing the following lines:

```
write ('Game Adapter Present');
write ('                    ');
if (equip_flag and game_mask) <> 0
then writeln ('YES')
else writeln ('NO');
```

and replacing them with these:

```
IF machine_id <> at_id THEN
  BEGIN
    write ('Game Adapter Present') ;
    write ('                    ');
    IF (equip_flag AND game_mask)
      < > 0 THEN writeln ('YES')
    ELSE
      writeln ('NO');
  END;
```

Finally, ATBIOS was changed to support I/O redirection so that a user can redirect the output to a file or printer. This change involves adding the following line:

```
{$P512}
```

anywhere before the first line of source code in ATBIOS.PAS.

ATKEY. ATKEY has not changed since it was introduced. This test works only with AT-compatible keyboards, not PC or XT keyboards. This is now the only one of the compatibility and performance tests that is specific to the AT.

ATPERF. Instead of a separate program that works only on 80386 machines, the new version of ATPERF is designed to run on a number of Intel processors. In fact, it can run on the 8088, 8086, 80286, and 80386, and it displays the type of processor contained in the computer. Changes also have been made to support zero-wait-state memory, such as that contained in IBM's XT-286. Like the original program, the new version is written in Microsoft C (ATPERF.C) with a MASM assembly language subroutine (TIME.ASM). Listings 1 and 2 contain the new source code for these routines.

In order to support several processors, ATPERF had to be modified so that it could detect the specific type of processor that is contained in the computer. The procedure called _CPU_TYPE that is in the TIME.ASM program performs part of this operation. The technique that is used in _CPU_TYPE is an Intel-approved method that is listed in the *Technical Reference Guide for Compaq's Deskpro 386.*

Compaq's procedure tests for the 8086/88, 80286, and 80386. It relies on the behavior of the FLAGS register in real mode, which differs among the processor types. The most significant bit of the FLAGS register (bit 15) distinguishes the 8086/88 from the other two types. In the 8086/88, bit 15 is always set to 1, whereas in the 80286 and 80386, its value is 0. Bits 12 through 14 are used to distinguish the 80286 from the 80386. In the 80286, these bits are always 0; however, the 80386 allows these bits to be set.

Armed with information about the Intel processor family, ATPERF invokes a procedure that is geared specifically toward that particular processor. Each of these procedures uses information about that processor (such as the number of processor cycles required to execute a particular instruction) when calculating the results of its tests. The procedure **measure88** tests the performance of computers that contain either the 8086 or 8088. The procedures **measure286** and **measure386** are called when the computer contains the 80286 or 80386, respectively.

When **measure88** runs, the results are used to distinguish between the 8- and 16-bit members of the family (between the 8088 and 8086). It does this by comparing the speed of two instructions: clear carry flag (CLC) and decimal adjust register AL after addition (DAA). CLC is a very fast, single-byte instruction, the speed of which is limited by the instruction fetch time. On an 8-bit computer, the execution speed of CLC is equal to the instruction fetch time. However, on 16-bit computers, CLC executes faster because instructions are fetched two at a time. DAA is a single-byte instruction that takes four CPU clock cycles to execute, about the same as an instruction fetch. Therefore, even though 16-bit computers can fetch two instructions at once, only one can be executed in that time. If the measurements for CLC and DAA are approximately equal, then the computer is of the 8-bit variety. If the CLC instruction takes less time, then the test is running on a 16-bit computer.

The **measure88** procedure performs a subset of the measurements performed by the 80286 and 80386 versions. It measures only the processor and numeric coprocessor speeds and the time for an instruction fetch (both byte and word). It performs these tests in the same way that the original version did. Processor speed is measured by executing a series of MUL instructions, each of which takes much longer than the number of cycles needed to fill the prefetch queue. Byte and word instruction fetches are measured by executing a series of CLC (byte) and MOV (word) instructions. These instructions take less time to execute than to fetch,

so this part of the test measures just the instruction fetching.

Measurements for memory read and write operations are not included in the **measure88** procedure. These operations cannot be measured accurately in the 8086 and 8088 processors because none of the memory reference instructions executes in less time than an instruction fetch.

The **measure286** and **measure386** procedures perform almost identical operations. The measurements they take are the same ones taken by the original ATPERF, with the addition of byte instruction fetches in both procedures and the 32-bit memory accesses that are included in **measure386**. The two procedures differ only in the assumptions each makes about the number of clock cycles that are required to execute certain instructions.

The mechanism that **measure286** and **measure386** use for measuring read and write times is slightly different than the method used in the earlier version of ATPERF. Previously, STO instructions were used to measure memory writes, and that value was then subtracted from a MOVS measurement (which includes both reading and writing) to obtain the value for memory read. The new version measures memory write with PUSHA instructions, which are much faster and can be used to test zero-wait-state memory. This change was made to handle the faster memory used in many of the new computers. STO instructions were limited to measuring one-wait-state memory or slower. With zero-wait-state memory, the STO instructions actually execute more slowly than their corresponding memory access, thereby invalidating the measurement that is obtained.

The Microsoft C compiler and Macro Assembler are used to produce the object code of the ATPERF program. Given that ATPERF.C contains the C source code and TIME.ASM contains the assembly language source code, the following commands are used to compile and link the program:

```
msc atperf.c;;
masm time.asm/r;;;
link atperf+time/map/stack:4000;;;;
```

The **/r** option on the **masm** command directs the assembler to generate floating-point instructions that can be executed on the numeric coprocessor. The **/stack:4000** option of the **link** command directs the linker to assign a stack of 4,000 bytes to the program. This option, along with the large stack it generates, is essential for running the new version of ATPERF. Because a series of PUSHA instructions is used to measure memory write operations, a large stack is necessary to contain the pushed values. The other option on the **link** command, **/map**, is not necessary for proper program execution; however, it does provide a link map that is quite helpful for debugging.

ATFLOAT. The ATFLOAT test has been updated to version 1.02 by correcting two lines of the program, both of which deal with displaying random characters to the screen. In the previous version of ATFLOAT, the time function, which always requires a pointer to a return value, used an uninitialized value for its return pointer. This sometimes caused random characters to appear. Version 1.02 replaces the line:

```
start = time();
```

with the line:

```
start = time(NULL);
```

and also replaces the line:

```
total = time() - start;
```

with:

```
total = time(NULL) - start;
```

ATDISK. ATDISK seemed to work fine when it was compiled under Microsoft C version 3.0. However, version 4.0 of the C compiler reported compile errors, because the program used structures when it should have used unions. The latest version of ATDISK (1.01) corrects the problem by replacing six instances of the line:

```
struct REGS regs;
```

with the line:

```
union REGS regs;
```

The updated compatibility and performance tests have performed their first job on the Compaq Deskpro 386, reviewed in this issue on p. 48. Because of the changes made to ATPERF, the program was able to identify the processor in the Deskpro and perform tests suitable for the 386.

Ted Forgeron, software project manager for Intel Scientific Computers, and Steven Armbrust, a technical writer, have reviewed several AT compatibles for PC Tech Journal. *With Paul Pierce, a software consultant, they devised the original AT Evaluation Suite.*

LISTING 1: ATPERF.C

```
/* ATPERF -- PC Tech Journal AT Hardware Performance Test
 * Version 2.00 Originally written 05/17/86
 * Last modified 12/18/86
 * Changes: 1. First release (1.00).
 *      2. Correct to work with zero-wait state memory (1.01).
 *      3. Detects an unsupported processor (1.01).
 *      4. Change to support 8088, 8086, 80188, 80186 and 80386 (2.00).
 * Copyright (c) 1986, 1987, Ziff Communications Company
 * Program by: Ted Forgeron, Paul Pierce
 * Measures clock rates and memory speeds
 * of PC and AT compatible computers. */

#define IRB     0       /* Instruction fetch, byte          */
#define IRW     1       /* Instruction fetch, word          */
#define IRX     2       /* Instruction fetch, slow byte */
#define MRB     3       /* RAM read byte                    */
#define MWB     4       /* RAM write byte                   */
#define RRB     5       /* ROM read byte                    */
#define ERB     6       /* EMM read byte                    */
#define EWB     7       /* EMM write byte                   */
#define VWB     8       /* Video write byte                 */
#define MRW     9       /* RAM read word                    */
#define MWW     10      /* RAM write word                   */
#define RRW     11      /* ROM read word                    */
#define ERW     12      /* EMM read word                    */
#define EWW     13      /* WMM write word                   */
#define VWW     14      /* Video write word                 */
#define IRD     15      /* Instruction fetch, dword         */
#define MRD     16      /* RAM read dword                   */
#define MWD     17      /* RAM write dword                  */
#define RRD     18      /* ROM read dword                   */
#define ERD     19      /* EMM read dword                   */
#define EWD     20      /* EMM write dword                  */
#define VWD     21      /* Video write dword                */
#define VARS88   3
#define VARS286 15
#define VARS386 22
/* Measurement procedures */
extern unsigned multime();
extern unsigned bclctime();
extern unsigned wmovtime();
extern unsigned dmovtime();
extern unsigned bdaatime();
extern unsigned bmvstime();
extern unsigned wmvstime();
extern unsigned dmvstime();
extern unsigned bromtime();
extern unsigned wromtime();
extern unsigned dromtime();
extern unsigned wpshtime();
extern unsigned dpshtime();
extern unsigned bemmtime();
extern unsigned wemptime();
```

We've taken the work out of doing Windows.

Microsoft® Windows is becoming the most popular operating environment for PC systems.

It's not surprising. Windows provides the foundation for an exciting new generation of applications that users are demanding. In addition, Windows handles many of the details involved in a software project allowing you to spend more time enhancing your application. That's why a growing number of corporate and independent software developers are building Windows applications.

The Microsoft Windows Software Development Kit is your key to this extraordinary new environment. It's packed with full reference documentation, libraries, utilities and sample programs. Together with our C Compiler or Macro Assembler, it's a comprehensive package that lets you make the most of your application.

Software with a new view.

Giving your applications the Windows treatment begins with a new look. The rich graphical environment allows you to rethink how you want your program to be presented on screen. It lets you mix text and graphics. You can incorporate multiple fonts in a variety of sizes, faces and styles. And it provides the basic building blocks that make it easy to create drop-down menus, dialog boxes, scroll bars, icons and more.

These features not only simplify your application design, but also provide the familiar interface that makes your software easier to learn and use.

Easing the data shuffle.

Of course, there's more to Windows than just looks. Now, different applications can work together. In concert.

The Windows Clipboard provides support for users to cut and paste information between your applications and others. Or you can use messages to establish "hot links" to transfer data automatically.

Upgrading made easy.

Windows' device independent design allows you to build your application today and take advantage of new technology as it becomes available. When new graphics cards, printers and pointing devices appear they can be used with your software, without modifying your code. Simply by installing the new driver.

Your window of opportunity.

The Microsoft Windows Software Development Kit is your fastest route to better applications. And with it, we also offer DIAL, our on-line technical support service to help you with the tough questions, and development courses that cover everything from using the dialog editor to memory management.

Find out how you can get your Microsoft Windows Software Development Kit. Pick up the phone and call (800) 426-9400. In Washington State and Alaska, call (206) 882-8088. In Canada, call (416) 673-7638. And we'll open the door to Windows.

The Microsoft Windows Software Development Kit includes:

- Dialog editor.
- Icon editor.
- Font editor.
- Resource compiler.
- Linker.
- MAKE (program maintenance utility).
- Symbolic debugger.
- Heap analysis utility.
- Sample programs.
- Windows libraries.
- Programmer's reference.
- Programmer's utility guide.

System requirements:

- 512K memory, DOS 2.0 or higher.
- Two double sided disk drives*
- Graphics adapter card.

*hard disk recommended

Microsoft® Windows Software Development Kit

The High Performance Software.

Microsoft is a registered trademark of Microsoft Corporation.

```
extern unsigned demptime();
extern unsigned wemmtime();
extern unsigned demmtime();
extern unsigned bvidtime();
extern unsigned wvidtime();
extern unsigned dvidtime();
extern unsigned fptime();
/* Timer rate in MHz */
#define TIMER2_RATE 1.193180
/* Number of processor clocks in a multiply instruction */
#define MULCLKS88 118
#define MULCLKS286 21
#define MULCLKS386 25
/* Overhead in the multiply test */
#define MULOVH88 ( 31 + 22*count/100 )
#define MULOVH286 ( 15 + 14*count/100 )
#define MULOVH386 ( 15 + 14*count/100 )
/* Overhead in the mov instruction test */
#define MOVOVH88 ( clktime * (31 + 22*count/100) )
#define MOVOVH286 ( clktime * (15 + 14*count/100) )
#define MOVOVH386 ( clktime * (15 + 14*count/100) )
/* Overhead in pusha instruction test */
#define WPOVH ( clktime * (15 + 9*count/200) )
#define DPOVH ( clktime * (15 + 9*count/200) )
/* Number of numeric processor clocks in a FP divide */
#define FPCLKS88 197
#define FPCLKS286 203
#define FPCLKS386 200
/* Processor overhead in the FP divide test */
#define FPOVH88 ( clktime * 2 * FPCOUNT )
#define FPOVH286 ( clktime * 9 * FPCOUNT )
#define FPOVH386 ( clktime * 9 * FPCOUNT )
/* Count for most tests */
#define COUNT 1000
/* Count for the f. p. divide test */
#define FPCOUNT 100
/* Number of trials for each test */
#define TRIALS 100
/* Variables which hold constants selected by cpu type */
int access_clks;
int cpu;        /* CPU type: 0=86/88,1=80186,2=80286,3=80386 */
double clkrate; /* Processor clock rate, MHz */
double clktime; /* Processor clock period, usec */
double fprate;  /* FP processor clock rate, MHz */
double fpacc;   /* FP processor clock period accumulator */
int emmok;      /* Set if extended memory is present */
int ndpok;      /* Set if math coprocessor is present */
int six;        /* Set if cpu is 8086 or 80186 */
double raw;     /* Variable for raw data */
double acctime[VARS386];   /* Accumulators for speeds */
int count;              /* Number of ops per trial */
int trials;             /* Number of repetitions */
/* Main program. */
main(argc, argv)
        int     argc;
        char    **argv;
{
        register int i;
        count = COUNT;
        trials = TRIALS;
/* Determine whether there a math coprocessor in the system. */
        ndpok = ndp_present();
/* Determine whether there is extended memory in the system and
 * allocate a piece of it for testing. */
        emmok = (setup_emm() == 0);
/* Detect the type of video card and save the information for the
 * video measurements. */
        setup_video();
/* Find out the CPU type and set parameters accordingly. */
        cpu = cpu_type();
        six = 0;
        switch (cpu) {
        case 0:
        case 1:
                access_clks = 4;
                measure88();
                break;
        case 2:
                access_clks = 2;
                measure286();
                break;
        case 3:
                access_clks = 2;
                measure386();
                break;
        default:
                printf("\nThis version of ATPERF is for ");
                printf("8088/86/188/186/286/386");
                printf("-based machines only.\n");
                exit(1);
        }
/* Release EMM memory page. */
        if (emmok)
                finish_emm();
/* Display the basic measurement results and performance index
 * relative to a 8 MHz AT. */
        printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
        printf("ATPERF -- PC Tech Journal AT Hardware ");
        printf("Performance Test\n");
        printf("Version 2.00, Copyright (c) 1986, 1987, Ziff ");
        printf("Communications Co.\n");
        printf("Written by Ted Forgeron and Paul Pierce\n");
        printf("IBM PC/AT model 339 (8 MHz) = 1.00 for relative ");
        printf("measurements.\n");
        if (cpu < 2)
                printf("\n");
        printf("                              ");
        printf("    Byte         Word         ");
        if (cpu > 2)
                printf("DWord    ");
        printf("Relative\n");
        printf("Average instruction fetch:");
        printf("%#10.2g uS", acctime[IRB]);
        printf("%#10.2g uS", acctime[IRW]);
        if (cpu > 2)
                printf("%#10.2g uS", acctime[IRD]);
        printf("%#10.2g\n", 0.403/acctime[IRW]);
        if (cpu < 2) {
                printf("\n");
        } else {
                printf("Average RAM read time:    ");
                printf("%#10.2g uS", acctime[MRB]);
                printf("%#10.2g uS", acctime[MRW]);
                if (cpu > 2)
                        printf("%#10.2g uS", acctime[MRD]);
                printf("%#10.2g\n", 0.401/acctime[MRW]);
                printf("Average RAM write time:   ");
                printf("%#10.2g uS", acctime[MWB]);
                printf("%#10.2g uS", acctime[MWW]);
                if (cpu > 2)
                        printf("%#10.2g uS", acctime[MWD]);
                printf("%#10.2g\n", 0.401/acctime[MWW]);
                if (emmok) {
                        printf("Average EMM read time:    ");
                        printf("%#10.2g uS", acctime[ERB]);
                        printf("%#10.2g uS", acctime[ERW]);
                        if (cpu > 2)
                                printf("%#10.2g uS", acctime[ERD]);
                        printf("%#10.2g\n", 0.402/acctime[ERW]);
                        printf("Average EMM write time:   ");
                        printf("%#10.2g uS", acctime[EWB]);
                        printf("%#10.2g uS", acctime[EWW]);
                        if (cpu > 2)
                                printf("%#10.2g uS", acctime[EWD]);
                        printf("%#10.2g\n", 0.402/acctime[EWW]);
                }
                printf("Average ROM read time:    ");
                printf("%#10.2g uS", acctime[RRB]);
                printf("%#10.2g uS", acctime[RRW]);
                if (cpu > 2)
                        printf("%#10.2g uS", acctime[RRD]);
                printf("%#10.2g\n", 0.401/acctime[RRW]);
                printf("Average Video write time: ");
                printf("%#10.2g uS", acctime[VWB]);
                printf("%#10.2g uS", acctime[VWW]);
                if (cpu > 2)
                        printf("%#10.2g uS", acctime[VWD]);
                printf("%#10.2g\n", 2.415/acctime[VWW]);
```

```
        }
        switch (cpu) {
        case 0:
                if (six)
                        printf(" 8086");
                else
                        printf(" 8088");
                break;
        case 1:
                if (six)
                        printf(" 80186");
                else
                        printf(" 80188");
                break;
        case 2:
                printf("80286");
                break;
        case 3:
                printf("80386");
                break;
        }
        printf(" CPU clock rate:        ");
        printf("%#4.1g MHz", clkrate);
        printf("  Relative: %#4.2g\n", clkrate/8.0);
        if (ndpok) {
                printf("Math Coprocessor clock rate: ");
                printf("%#4.1g MHz", fprate);
                printf("  Relative: %#4.2g\n", fprate/5.33);
        }
/* Calculate refresh overhead from instruction fetch time by assuming
 * that each fetch takes an exact multiple of the clock period.  The
 * difference between average time and the time for an individual
 * fetch is due to memory refresh cycles. */
        raw = acctime[IRW] / clktime;
        printf("Refresh overhead:           %#2.1g%%\n",
                ( (raw - (int)raw) / (int)raw ) * 100);
/* Print information about memory based on the speed measurements. */
        printf("\nMemory   ");
        printf("       Access width          Wait states\n");
        if (cpu < 2) {
                if (six)
                        analyze("Fetch   ", acctime[IRW],
                                acctime[IRW], 2.0*acctime[IRW]);
                else
                        analyze("Fetch   ", acctime[IRB],
                                acctime[IRW], 2.0*acctime[IRW]);
        } else {
                analyze("RAM read ",
                        acctime[MRB], acctime[MRW], acctime[MRD]);
                analyze("RAM write",
                        acctime[MWB], acctime[MWW], acctime[MWD]);
                if (emmok) {
                        analyze("EMM read",
                         acctime[ERB], acctime[ERW], acctime[ERD]);
                        analyze("EMM write",
                         acctime[EWB], acctime[EWW], acctime[EWD]);
                }
                analyze("ROM read",
                        acctime[RRB], acctime[RRW], acctime[RRD]);
                analyze("Video write",
                        acctime[VWB], acctime[VWW], acctime[VWD]);
        }
}
measure88()
{
        register int i;
/* Measure the clock rate by executing multiply instructions.  Each
 * multiply takes a fixed number of clock cycles. */
        clktime = 0;
        for (i = 0; i < trials; i++) {
/* Obtain the number of clock ticks for "count" multiplies. */
                raw = multime(count);
/* Accumulate the clock time in microseconds by adjusting for the
 * timer rate, number of clocks per multiply, instruction count, and
 * test overhead. */
                clktime +=  raw / (TIMER2_RATE *
                        ((double)MULCLKS88*count + MULOVH88));
        }
/* Calculate the average clock period by dividing by the number of
 * trials.  The clock rate is the inverse of the clock period. */
        clktime /= trials;
        clkrate = 1.0/clktime;
/* Clear all of the memory speed accumulators. */
        for (i = 0; i < VARS88; i++)
                acctime[i] = 0;
/* Do the memory speed tests. */
        for (i = 0; i < trials; i++) {
/* Obtain the number of timer ticks for "count" clc instructions,
 * which are limited by memory fetch time. */
                raw = bclctime(count);
/* Accumulate the number of microseconds per instruction fetch by
 * adjusting for the timer rate, test overhead, and instruction
 * count. */
                acctime[IRB] +=
                        (raw / TIMER2_RATE - MOVOVH88) / count;
/* Make a similar measurement for the two byte "mov" instruction. */
                raw = wmovtime(count);
                acctime[IRW] +=
                        (raw / TIMER2_RATE - MOVOVH88) / count;
/* Make a similar measurement for the 4 clock "daa" instruction. */
                raw = bdaatime(count);
                acctime[IRX] +=
                        (raw / TIMER2_RATE - MOVOVH88) / count;
        }
/* Calculate averages for all measurements. */
        for (i = 0; i < VARS88; i++)
                acctime[i] /= trials;
/* Calculate numeric processor clock rate using floating point divide
 * instructions, using the same technique as was used to measure
 * the processor clock rate. */
        if (ndpok) {
                fprate = 0;
                for (i = 0; i < trials; i++) {
                        raw = fptime(FPCOUNT);
                        fpacc +=  (raw / TIMER2_RATE - FPOVH88) /
                                FPCLKS88 / FPCOUNT;
                }
                fpacc /= trials;
                fprate = 1.0/fpacc;
        }
/* Set 86 flag if 2 clock byte instructions execute faster than 4
 * clock instructions due to being fetched two to a word. */
        six = acctime[IRB] < 0.9*acctime[IRX];
}
measure286()
{
        register int i;
/* Measure the clock rate by executing multiply instructions.  Each
 * multiply takes a fixed number of clock cycles. */
        clktime = 0;
        for (i = 0; i < trials; i++) {
/* Obtain the number of clock ticks for "count" multiplies. */
                raw = multime(count);
/* Accumulate the clock time in microseconds by adjusting for the
 * timer rate, number of clocks per multiply, instruction count, and
 * test overhead. */
                clktime +=  raw / (TIMER2_RATE *
                        ((double)MULCLKS286*count + MULOVH286));
        }
/* Calculate the average clock period by dividing by the number of
 * trials.  The clock rate is the inverse of the clock period. */
        clktime /= trials;
        clkrate = 1.0/clktime;
/* Clear all of the memory speed accumulators. */
        for (i = 0; i < VARS286; i++)
                acctime[i] = 0;
/* Do the memory speed tests. */
        for (i = 0; i < trials; i++) {
/* Obtain the number of timer ticks for "count" clc instructions,
 * which are limited by memory fetch time. */
                raw = bclctime(count);
/* Accumulate the number of microseconds per instruction fetch by
 * adjusting for timer rate, test overhead, and instruction count. */
                acctime[IRB] +=
                        (raw / TIMER2_RATE - MOVOVH286) / count;
/* Make a similar measurement for the two byte "mov" instruction. */
                raw = wmovtime(count);
                acctime[IRW] +=
```

```
                        (raw / TIMER2_RATE - MOVOVH286) / count;
/* Measure byte read+write time measuring movs instructions. */
                raw = bmvstime(count);
                acctime[MRB] += raw/(TIMER2_RATE*count);
/* Calculate ROM read time by measuring movs from ROM to RAM. */
                raw = bromtime(count);
                acctime[RRB] += raw/(TIMER2_RATE*count);
/* Measure word write using the pusha instruction. */
                raw = wpshtime(count) - WPOVH;
                acctime[MWW] += raw/(TIMER2_RATE*count);
/* Measure movs (read+write) time. */
                raw = wmvstime(count);
                acctime[MRW] += raw/(TIMER2_RATE*count);
                raw = wromtime(count);
                acctime[RRW] += raw/(TIMER2_RATE*count);
/* If EMM is present, do measurements on it using same techniques. */
                if (emmok) {
/* Measure byte mov in EMM. */
                        raw = bemmtime(count);
                        acctime[ERB] += raw/(TIMER2_RATE*count);
/* Measure word write, calculate word read. */
                        raw = wemptime(count) - WPOVH;
                        acctime[EWW] += raw/(TIMER2_RATE*count);
                        raw = wemmtime(count);
                        acctime[ERW] += raw/(TIMER2_RATE*count);
                }
/* Measure byte and word writes into video RAM. */
                raw = bvidtime(count);
                acctime[VWB] += raw/(TIMER2_RATE*count);
                raw = wvidtime(count);
                acctime[VWW] += raw/(TIMER2_RATE*count);
        }
/* Calculate averages for all measurements. */
        for (i = 0; i < VARS286; i++)
                acctime[i] /= trials;
/* Adjust word write times by subtracting instruction fetch time. */
        acctime[MWW] -= acctime[IRW]/16;
        if (emmok)
                acctime[EWW] -= acctime[IRW]/16;
/* Adjust for extra time per instruction when measuring zero wait
 * state memory. */
        if (acctime[MWW] < 3.375*clktime)
                acctime[MWW] -= clktime/8;
        if (emmok)
                if (acctime[EWW] < 3.375*clktime)
                        acctime[EWW] -= clktime/8;
/* Calculate byte write time by assuming the same ratio between read
 * and write as for word access. */
        acctime[MWB] = acctime[MRB] * acctime[MWW] /
                        acctime[MRW];
        if (emmok)
                acctime[EWB] = acctime[ERB] * acctime[EWW] /
                                acctime[ERW];
/* Calculate read times by subtracting write time from
 * mov (read+write) time. */
        acctime[MRB] = acctime[MRB] - acctime[MWB];
        acctime[MRW] = acctime[MRW] - acctime[MWW];
        acctime[RRB] = acctime[RRB] - acctime[MWB];
        acctime[RRW] = acctime[RRW] - acctime[MWW];
        if (emmok) {
                acctime[ERB] = acctime[ERB] - acctime[EWB];
                acctime[ERW] = acctime[ERW] - acctime[EWW];
        }
/* Calculate numeric processor clock rate using floating point divide
 * instructions, using the same technique as was used to measure the
 * processor clock rate. */
        if (ndpok) {
                fprate = 0;
                for (i = 0; i < trials; i++) {
                        raw = fptime(FPCOUNT);
                        fpacc +=  (raw / TIMER2_RATE - FPOVH286) /
                                FPCLKS286 / FPCOUNT;
                }
                fpacc /= trials;
                fprate = 1.0/fpacc;
        }
/* Fill in dword variables to provide complete input to analyze. */
        acctime[MRD] = 2.0 * acctime[MRW];
        acctime[MWD] = 2.0 * acctime[MWW];
        acctime[ERD] = 2.0 * acctime[ERW];
        acctime[EWD] = 2.0 * acctime[EWW];
        acctime[RRD] = 2.0 * acctime[RRW];
        acctime[VWD] = 2.0 * acctime[VWW];
}
measure386()
{
        register int i;
/* Measure the clock rate by executing multiply instructions.  Each
 * multiply takes a fixed number of clock cycles. */
        clktime = 0;
        for (i = 0; i < trials; i++) {
/* Obtain the number of clock ticks for "count" multiplies. */
                raw = multime(count);
/* Accumulate the clock time in microseconds by adjusting for the
 * timer rate, number of clocks per multiply, instruction count, and
 * test overhead. */
                clktime +=  raw / (TIMER2_RATE *
                        ((double)MULCLKS386*count + MULOVH386));
        }
/* Calculate the average clock period by dividing by the number of
 * trials.  The clock rate is the inverse of the clock period. */
        clktime /= trials;
        clkrate = 1.0/clktime;
/* Clear all of the memory speed accumulators. */
        for (i = 0; i < VARS386; i++)
                acctime[i] = 0;
/* Do the memory speed tests. */
        for (i = 0; i < trials; i++) {
/* Obtain the number of timer ticks for "count" clc instructions,
 * which are limited by memory fetch time. */
                raw = bclctime(count);
/* Accumulate the number of microseconds per instruction fetch by
 * adjusting for timer rate, test overhead, and instruction count. */
                acctime[IRB] +=
                        (raw / TIMER2_RATE - MOVOVH386) / count;
/* Make a similar measurement for the two byte and four byte "mov"
 * instruction. */
                raw = wmovtime(count);
                acctime[IRW] +=
                        (raw / TIMER2_RATE - MOVOVH386) / count;
                raw = dmovtime(count);
                acctime[IRD] +=
                        (raw / TIMER2_RATE - MOVOVH386) / count;
/* Measure byte read+write time measuring movs instructions. */
                raw = bmvstime(count);
                acctime[MRB] += raw/(TIMER2_RATE*count);
/* Calculate ROM read time by measuring movs from ROM to RAM. */
                raw = bromtime(count);
                acctime[RRB] += raw/(TIMER2_RATE*count);
/* Measure word and dword write using the pusha instruction. */
                raw = wpshtime(count) - WPOVH;
                acctime[MWW] += raw/(TIMER2_RATE*count);
                raw = dpshtime(count) - DPOVH;
                acctime[MWD] += raw/(TIMER2_RATE*count);
/* Measure word and dword movs (read+write) time. */
                raw = wmvstime(count);
                acctime[MRW] += raw/(TIMER2_RATE*count);
                raw = wromtime(count);
                acctime[RRW] += raw/(TIMER2_RATE*count);
                raw = dmvstime(count);
                acctime[MRD] += raw/(TIMER2_RATE*count);
                raw = dromtime(count);
                acctime[RRD] += raw/(TIMER2_RATE*count);
/* If EMM is present, do measurements on it using same techniques. */
                if (emmok) {
                        raw = bemmtime(count);
                        acctime[ERB] += raw/(TIMER2_RATE*count);
                        raw = wemptime(count) - WPOVH;
                        acctime[EWW] += raw/(TIMER2_RATE*count);
                        raw = wemmtime(count);
                        acctime[ERW] += raw/(TIMER2_RATE*count);
                        raw = demptime(count) - DPOVH;
                        acctime[EWD] += raw/(TIMER2_RATE*count);
                        raw = demmtime(count);
                        acctime[ERD] += raw/(TIMER2_RATE*count);
                }
/* Measure writes into video RAM. */
                raw = bvidtime(count);
```

```
                acctime[VWB] += raw/(TIMER2_RATE*count);
                raw = wvidtime(count);
                acctime[VWW] += raw/(TIMER2_RATE*count);
                raw = dvidtime(count);
                acctime[VWD] += raw/(TIMER2_RATE*count);
        }
/* Calculate averages for all measurements. */
        for (i = 0; i < VARS386; i++)
                acctime[i] /= trials;
/* Adjust word write times by subtracting instruction fetch time. */
        acctime[MWW] -= acctime[IRW]/16;
        acctime[MWD] -= acctime[IRW]/8;
        if (emmok) {
                acctime[EWW] -= acctime[IRW]/16;
                acctime[EWD] -= acctime[IRW]/8;
        }
/* Adjust for extra time per instruction when measuring zero wait
 * state memory. */
        if (acctime[MWW] < 3.375*clktime)
                acctime[MWW] -= clktime;
        if (acctime[MWD] < 3.375*clktime)
                acctime[MWD] -= clktime;
        if (emmok) {
                if (acctime[EWW] < 3.375*clktime)
                        acctime[EWW] -= clktime;
                if (acctime[EWD] < 3.375*clktime)
                        acctime[EWD] -= clktime;
        }
/* Calculate byte write time by assuming the same ratio between read
 * and write as for word access. */
        acctime[MWB] = acctime[MRB] * acctime[MWW] /
                       acctime[MRW];
        if (emmok)
                acctime[EWB] = acctime[ERB] * acctime[EWW] /
                               acctime[ERW];
/* Calculate read times by subtracting write time from
 * mov (read+write) time. */
        acctime[MRB] = acctime[MRB] - acctime[MWB];
        acctime[MRW] = acctime[MRW] - acctime[MWW];
        acctime[MRD] = acctime[MRD] - acctime[MWD];
        acctime[RRB] = acctime[RRB] - acctime[MWB];
        acctime[RRW] = acctime[RRW] - acctime[MWW];
        acctime[RRD] = acctime[RRD] - acctime[MWD];
        if (emmok) {
                acctime[ERB] = acctime[ERB] - acctime[EWB];
                acctime[ERW] = acctime[ERW] - acctime[EWW];
                acctime[ERD] = acctime[ERD] - acctime[EWD];
        }
/* Calculate numeric processor clock rate using floating point divide
 * instructions, using the same technique as was used to measure the
 * processor clock rate. */
        if (ndpok) {
                fprate = 0;
                for (i = 0; i < trials; i++) {
                        raw = fptime(FPCOUNT);
                        fpacc +=  (raw / TIMER2_RATE - FPOVH386) /
                                FPCLKS386 / FPCOUNT;
                }
                fpacc /= trials;
                fprate = 1.0/fpacc;
        }
}
/* analyze. This procedure deduces information about the memory based
 * on the measured times. If byte (8 bits) and word (16 bits) times
 * are different then the memory is byte oriented since each word
 * operation takes two byte operations.  Otherwise, if the byte and
 * word times are about the same, the memory is word oriented and can
 * access either a word or a byte in a single memory cycle.
 * Similar arguments can be made about 32 bit accesses.
 * Each memory access takes an exact number of processor clock
 * cycles.  The first two are required by the processor, but
 * any additional cycles are determined by the memory and are
 * called wait states (because the processor is waiting for
 * the memory.) */
analyze(name, btime, wtime, dtime)
        char    *name;
        double  btime;
        double  wtime;
        double  dtime;
{
        double  t;
/* Print the heading */
        printf("%-12s", name);
/* Determine whether the memory is byte oriented, word oriented, dword
 * oriented, or other. (If other, the data are suspect.) */
        if (wtime > dtime*0.66 &&
            wtime < dtime*1.33) {
                printf("        Dword ");
                t = dtime;
        } else if (wtime*2 > dtime*0.66 &&
                   wtime*2 < dtime*1.33 &&
                   btime > wtime*0.66 &&
                   btime < wtime*1.33) {
                printf("        Word  ");
                t = wtime;
        } else if (btime*2 > wtime*0.66 &&
                   btime*2 < wtime*1.33) {
                printf("        Byte  ");
                t = btime;
        } else {
                printf("       Strange");
                t = btime;
        }
/* Determine the number of wait states by dividing by the clock
 * period, subtracting two processor clock times, and rounding down
 * to an integer. */
        t = t / clktime - access_clks;
        if (t < 0.0)
                t = 0.0;
        printf("              %6d\n", (unsigned)t);
}
```

LISTING 2: TIME.ASM

```
                NAME    TIME

_TEXT   SEGMENT  PARA PUBLIC 'CODE'
_TEXT   ENDS
CONST   SEGMENT  PARA PUBLIC 'CONST'
CONST   ENDS
_BSS    SEGMENT  PARA PUBLIC 'BSS'
_BSS    ENDS
_DATA   SEGMENT  PARA PUBLIC 'DATA'
_DATA   ENDS
DGROUP  GROUP   CONST,  _BSS,   _DATA
        ASSUME  CS: _TEXT, DS: DGROUP, SS: DGROUP, ES: DGROUP
TESTSEG SEGMENT PARA PUBLIC 'TEST'
TESTSEG_START   DW      32767 DUP (?)
TESTSEG ENDS
PPI_PORT        EQU     061H
TIMER2_PORT     EQU     042H
TIMER_CTRL      EQU     043H
_DATA           SEGMENT
VIDBASE         DW      0B800H
EMMBASE         DW      9000H
PID             DW      ?
EMM_NAME        DB      "EMMXXXX0"
_DATA           ENDS
_TEXT           SEGMENT
;*************************************************************;
;       _MULTIME                                              ;
;       TIME EXECUTION OF MULTIPLY INSTRUCTIONS               ;
;*************************************************************;
        EVEN
        DB      ?, ?
                PUBLIC  _MULTIME
_MULTIME        PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 08000H              ; SET DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        NOP                             ; ALIGN INSTRUCTIONS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
```

```
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
ML:     REPT    100                     ; DO 100 MULTIPLIES
        MUL     DI                      ;
        ENDM                            ; END MACRO
        DEC     CX                      ; COUNT THIS PASS
        JZ      MD                      ; JUMP IF COMPLETE
        JMP     ML                      ; LOOP BACK IF NOT DONE
MD:     MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_MULTIME        ENDP
;*************************************************************;
;       _WMOVTIME                                             ;
;       TIME EXECUTION OF MOV INSTRUCTION (INSTR. READ TIME)    ;
;*************************************************************;
        EVEN
                PUBLIC  _WMOVTIME
_WMOVTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        NOP                             ; ALIGN INSTRUCTIONS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
IL:     REPT    100                     ; DO 100 MOVES
        MOV     DX, BX                  ;
        ENDM                            ; END MACRO
        DEC     CX                      ; COUNT THIS PASS
        JZ      ID                      ; JUMP IF COMPLETE
        JMP     IL                      ; LOOP BACK IF NOT DONE
ID:     MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_WMOVTIME       ENDP
;*************************************************************;
;       _DMOVTIME                                             ;
;       TIME EXECUTION OF MOV INSTRUCTION (INSTR. READ TIME)    ;
;*************************************************************;
        EVEN
        DB      ?, ?
        DB      0, 0
                PUBLIC  _DMOVTIME
_DMOVTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        NOP                             ; ALIGN INSTRUCTIONS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
NL:     REPT    100                     ; DO 100 MOV DX,05555H
        DB      0C7H, 0C2H, 055H, 055H  ;  THE LONG WAY (WITH MOD R/M)
        ENDM                            ; END MACRO
        DEC     CX                      ; COUNT THIS PASS
        JZ      ND                      ; JUMP IF COMPLETE
        JMP     NL                      ; LOOP BACK IF NOT DONE
ND:     MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_DMOVTIME       ENDP
;*************************************************************;
;       _BCLCTIME                                             ;
;       TIME EXECUTION OF CLC INSTRUCTION (INSTR. READ TIME)    ;
;*************************************************************;
        EVEN
                PUBLIC  _BCLCTIME
_BCLCTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        NOP                             ; ALIGN INSTRUCTIONS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
BIL:    REPT    100                     ; DO 100 CLC'S
        CLC                             ;
        ENDM                            ; END MACRO
        DEC     CX                      ; COUNT THIS PASS
        JZ      BID                     ; JUMP IF COMPLETE
        JMP     BIL                     ; LOOP BACK IF NOT DONE
BID:    MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_BCLCTIME       ENDP
;*************************************************************;
;       _BDAATIME                                             ;
;       TIME EXECUTION OF DAA INSTRUCTION (INSTR. READ TIME)    ;
;*************************************************************;
        EVEN
        DB      ?, ?
                PUBLIC  _BDAATIME
_BDAATIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        NOP                             ; ALIGN INSTRUCTIONS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
BXL:    REPT    100                     ; DO 100 DAA'S
        DAA                             ;
        ENDM                            ; END MACRO
        DEC     CX                      ; COUNT THIS PASS
        JZ      BXD                     ; JUMP IF COMPLETE
```

FORTRAN PROGRAMMERS

LCS ANNOUNCES F77L LAHEY FORTRAN VERSION 2.20
WE JUST MADE OUR TOP RATED FORTRAN LANGUAGE SYSTEM BETTER.

"Lahey's F77L FORTRAN is the compiler of choice. It's definitely a 'Programmer's FORTRAN,' with features to aid both the casual and the professional programmer . . . F77L compiled the five files in a total of 12 minutes, which was 4 times as fast as MS FORTRAN and an astounding 6 times as fast as Pro FORTRAN..."

—Editor's Choice *PC Magazine*

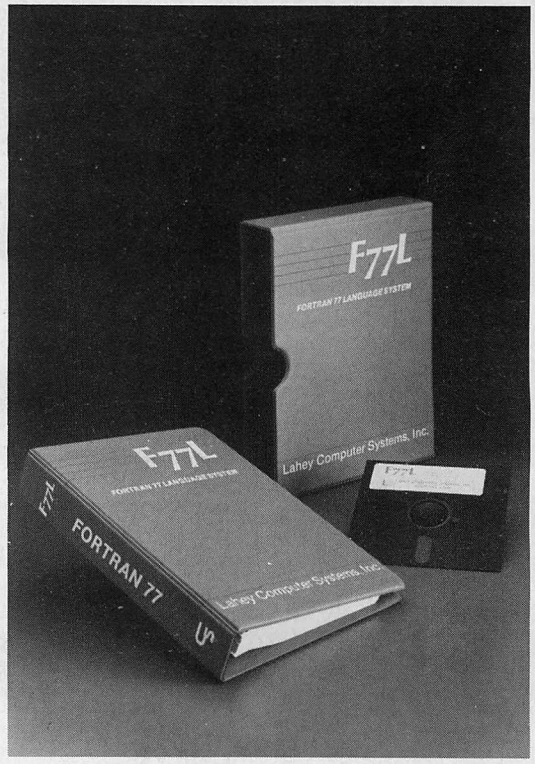
HERE ARE JUST A FEW OF THE REASONS WHY F77L IS THE COMPILER OF CHOICE:

- Full Implementation of the ANSI 77 Standard
- Fast Compilation—outruns everything on the market
- Powerful Multi-Featured Source On-Line Debugger
- Popular Extensions for easy porting of mainframe and mini computer programs (Including NAMELIST)
- Recursion—allocates local variables on the stack
- Arrays and COMMONS greater than 64K
- Clear and Precise English Diagnostics
- Long Variable Names—Up to 31 Characters
- COMPLEX*16, LOGICAL*1 and INTEGER*2
- IEEE Standard Floating Point
- Compatibility with popular third party software
- Unmatched Technical Support with an on-line bulletin board

NEW FEATURES WITH VERSION 2.20:

- Cross reference and source listings
- Allocation maps of COMMON variables and arrays
- In-line comments
- IMPLICIT NONE compiler option
- Faster Execution
- Source On-Line Debugger (SOLD) includes: Trace Execution; No Relinking required; On-screen Listing; No effect on code size

Call about our New F77L development tools:
Lahey Profiler Mathematical Functions Library Overlay Linker



It is more than just features that make F77L an outstanding product; it is the years of experience behind the software. At Lahey Computer Systems, we have been developing FORTRAN compilers since 1967 and we are committed to keeping F77L the industry leader.

When *PC Magazine* selected our version 1.35 as the Editor's Choice among PC FORTRANs, we were pleased but not completely satisfied—we knew we could improve the product. F77L Version 2.20 increases our lead over the competition. F77L's precise diagnostics, advanced debug package, helpful user screens and comprehensive manual make it a complete and easy to use high productivity tool.

When evaluating any software package, an important factor to consider is the value of your time. F77L saves you time and money the moment you start using it. Our FORTRAN Language System has the key features you need to increase productivity and get the job done. Other PC FORTRANs may be cheaper than F77L, but none are less expensive to use.

F77L—THE PROGRAMMER'S FORTRAN

Price: $477.00 Includes Free UPS 2nd Day shipping.
System Requirements: MS-DOS or PC-DOS (2.0 or greater), 256K, math coprocessor (8087-80287)

TO ORDER OR FOR MORE INFORMATION:
702-831-2500

Lahey Computer Systems, Inc.
P.O. Box 6091, Incline Village, NV 89450
Telex: 9102401256

International Representatives:
Canada: Barry Mooney & Assoc., Tel. (902)6652941 • **England:** Grey Matter Ltd, Tel. (0364)53499 • **Switzerland:** DST Comp. Services, Tel. (022)989188
Denmark: Ravenholm Computing, Tel. (02)887249 • **Australia:** Comp. Transitions, Tel. (03)5372786 • **Japan:** Microsoftware Inc., Tel. (03)8138222

MS-DOS & MS FORTRAN are trademarks of Microsoft Corporation. Pro FORTRAN refers to IBM PC Professional FORTRAN by Ryan McFarland

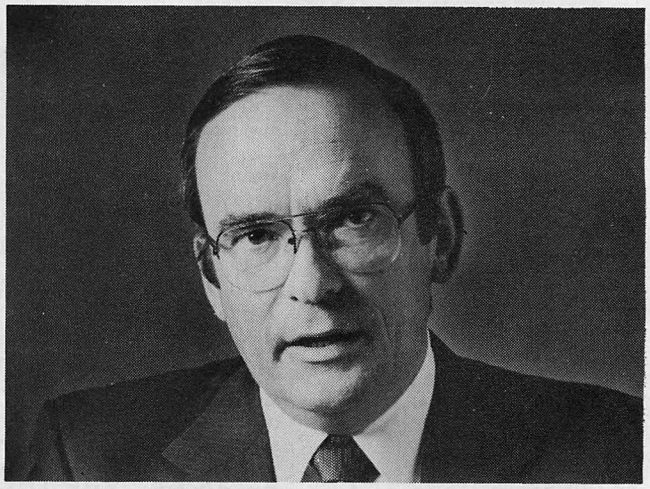
```
        JMP     BXL                     ; LOOP BACK IF NOT DONE
BXD:    MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_BDAATIME       ENDP
;*************************************************************;
;       _BMVSTIME                                             ;
;       TIME EXECUTION OF REP MOVSB INSTRUCTION               ;
;*************************************************************;
                PUBLIC  _BMVSTIME
_BMVSTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        LEA     SI, TESTSEG_START       ; DS:SI -> TEST SEGMENT
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_BMVSTIME       ENDP
;*************************************************************;
;       _WMVSTIME                                             ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION               ;
;*************************************************************;
                PUBLIC  _WMVSTIME
_WMVSTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        LEA     SI, TESTSEG_START       ; DS:SI -> TEST SEGMENT
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_WMVSTIME       ENDP
;*************************************************************;
;       _DMVSTIME                                             ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION               ;
;*************************************************************;
                PUBLIC  _DMVSTIME
_DMVSTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     DI, DI                  ; CLEAR EDI
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     SI, SI                  ; CLEAR ESI
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     CX, CX                  ; CLEAR ECX
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        LEA     SI, TESTSEG_START       ; DS:SI -> TEST SEGMENT
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_DMVSTIME       ENDP
;*************************************************************;
;       _WPSHTIME                                             ;
;       TIME EXECUTION OF PUSHA INSTRUCTION                   ;
;*************************************************************;
        EVEN
        DB      ?
                PUBLIC  _WPSHTIME
_WPSHTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        CWD                             ; MAKE DOUBLE WORD
        MOV     CX, 200                 ;
        DIV     CX                      ; DIVIDE BY MOVS/LOOP
        MOV     CX, AX                  ;
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
WSL:    REPT    25                      ; PUSH THE REGISTERS
        DB      60H                     ;
        ENDM                            ; END MACRO
        MOV     SP, BP                  ; PUT THE STACK BACK
        LOOP    WSL                     ; LOOP UNTIL DONE
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        MOV     SP, BP                  ; PUT THE STACK BACK
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
```

STRAIGHT TALK

Our NETBIOS LAN/Gateway

...has been designed by our professionals to meet the most exacting standards. If your installation has a unique condition, just call us. Our technicians in Scottsdale will fix it, or under our QuickFix Program will be on your doorstep within 48 hours...probably much sooner. And if they can't make it, call me, and I will.*

Jim Pitre,
President



With LinkUp products your beginning purchase is just that...a foundation that works.

Doesn't it make ever so much more sense to develop your software and computer systems as upgradable assets? Begin with our LinkUp 3270 Interactive Terminal Facility (ITF) connection to the mainframe, then expand or change as your needs require to include...

...File Transfer. Two programs for your consideration. One interfaces with the TSO or CMS editor; one works with IBM 3270 PC host-based file transfer under TSO, CMS or CICS...

...or **Application Program Interface.** Integrate your own programs to use the capabilities of 3270 SNA/BSC, 3770 SNA and 3780 BSC emulations...

...or **LinkUp SpeedPrint.** Data rates up to 19.2 Kbps from an application or disk to a dedicated high-speed printer, eliminating host/local print conflict.

...or **LAN/Gateway.** With the addition of interfacing modules the LinkUp 3270 ITF runs on most Local Area Networks. Use the advanced package for datacom between PCs on a LAN and IBM or compatible host.

These intelligent software capabilities offer the options your computer environment just may require. For today. For tomorrow.

***QuickFix Program**

In addition to telephone assistance from Scottsdale, Information Technologies, Inc. stands ready to help you in critical situations. We're different because when our product is at fault and it can't be fixed from Scottsdale, we will come to you anywhere in the contiguous 48 states at no charge. If the problem is caused by your PC, the way you've configured it, the mainframe, or any reason other than our product, then you pay our standard consulting fee plus expenses.

Designed, engineered and manufactured in the U.S.A.

Advanced micro-to-mainframe solutions

INFORMATION TECHNOLOGIES INC.®

7850 East Evans Road, Scottsdale, Arizona 85260 1-800-431-3460 1-602-998-1033

Call us. Our engineers speak SNA/SDLC, BSC, ASCII, EBCDIC and Human.

CIRCLE NO. 138 ON READER SERVICE CARD

```
_WPSHTIME       ENDP
;*************************************************************;
;       _DPSHTIME                                             ;
;       TIME EXECUTION OF PUSHA INSTRUCTION                   ;
;*************************************************************;
        EVEN
        DB      ?, ?, ?
                PUBLIC  _DPSHTIME
_DPSHTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        CWD                             ; MAKE DOUBLE WORD
        MOV     CX, 200                 ;
        DIV     CX                      ; DIVIDE BY MOVS/LOOP
        MOV     CX, AX                  ;
        AND     SP, 0FFFCH              ; ALIGN SP
        PUSH    AX                      ; DUMMY
        PUSH    BP                      ; SAVE BP
        MOV     BP, SP                  ;
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
DSL:    REPT    25                      ; PUSH THE BIG REGISTERS
        DB      66H, 60H                ;
        ENDM                            ; END MACRO
        MOV     SP, BP                  ; PUT THE STACK BACK
        LOOP    DSL                     ; LOOP UNTIL DONE
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        POP     BP                      ;
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        MOV     SP, BP                  ; PUT THE STACK BACK
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_DPSHTIME       ENDP
;*************************************************************;
;       _BROMTIME                                             ;
;       TIME EXECUTION OF REP MOVSB INSTRUCTION FROM ROM      ;
;*************************************************************;
                PUBLIC  _BROMTIME
_BROMTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DI, 0F000H              ; SET DS TO ROM START
        MOV     DS, DI                  ;
        MOV     SI, 0                   ; DS:SI -> ROM
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_BROMTIME       ENDP
```

```
;*************************************************************;
;       _WROMTIME                                             ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION FROM ROM      ;
;*************************************************************;
                PUBLIC  _WROMTIME
_WROMTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DI, 0F000H              ; SET DS TO ROM START
        MOV     DS, DI                  ;
        MOV     SI, 0                   ; DS:SI -> ROM
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_WROMTIME       ENDP
;*************************************************************;
;       _DROMTIME                                             ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION FROM ROM      ;
;*************************************************************;
                PUBLIC  _DROMTIME
_DROMTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     DI, DI                  ; CLEAR EDI
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     SI, SI                  ; CLEAR ESI
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     CX, CX                  ; CLEAR ECX
        MOV     DI, TESTSEG             ;
        MOV     ES, DI                  ;
        MOV     DI, 0F000H              ; SET DS TO ROM START
        MOV     DS, DI                  ;
        MOV     SI, 0                   ; DS:SI -> ROM
        LEA     DI, TESTSEG_START       ; ES:DI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
```

```
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_DROMTIME       ENDP
;*************************************************************;
;       _BVIDTIME                                             ;
;       TIME EXECUTION OF REP STOSB INTO VIDEO MEMORY         ;
;*************************************************************;
                PUBLIC  _BVIDTIME
_BVIDTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     AX, VIDBASE             ; GET BASE ADDRESS
        MOV     ES, AX                  ;
        MOV     DI, 0                   ; ES:DI -> VIDEO MEMORY
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_BVIDTIME       ENDP
;*************************************************************;
;       _WVIDTIME                                             ;
;       TIME EXECUTION OF REP STOSW INTO VIDEO MEMORY         ;
;*************************************************************;
                PUBLIC  _WVIDTIME
_WVIDTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     AX, VIDBASE             ; GET BASE ADDRESS
        MOV     ES, AX                  ;
        MOV     DI, 0                   ; ES:DI -> VIDEO MEMORY
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP STOSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_WVIDTIME       ENDP
;*************************************************************;
;       _DVIDTIME                                             ;
;       TIME EXECUTION OF REP STOSW INTO VIDEO MEMORY         ;
;*************************************************************;
                PUBLIC  _DVIDTIME
_DVIDTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    ES                      ; SAVE ES
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     DI, DI                  ; CLEAR EDI
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     CX, CX                  ; CLEAR ECX
        MOV     AX, VIDBASE             ; GET BASE ADDRESS
        MOV     ES, AX                  ;
        DB      066H                    ; 32 BIT OPERANDS:
        MOV     AX, 0700H               ;  MOV EAX, 07000700H
        DW      0700H                   ;
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        REP STOSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     ES                      ; RESTORE ES
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_DVIDTIME       ENDP
;*************************************************************;
;       _SETUP_VIDEO                                          ;
;       DETECT THE TYPE OF VIDEO CARD AND SAVE THE BASE       ;
;*************************************************************;
                PUBLIC  _SETUP_VIDEO
_SETUP_VIDEO    PROC    NEAR
        PUSH    BP                      ; SAVE REGISTERS
        PUSH    ES                      ;
        PUSH    SI                      ;
        PUSH    DI                      ;
        INT     11H                     ; EQUIPMENT DETERMINATION
        AND     AL, 30H                 ; MASK DISPLAY BITS
        CMP     AL, 30H                 ; CHECK FOR MONOCHROME
        MOV     AX, 0B000H              ; MONOCHROME BASE
        JE      SVM                     ; JUMP IF MONOCHROME
        MOV     AX, 0B800H              ; COLOR BASE
SVM:    MOV     VIDBASE, AX             ; SAVE BASE ADDRESS
        POP     DI                      ; RESTORE REGISTERS
        POP     SI                      ;
        POP     ES                      ;
        POP     BP                      ;
        RET                             ; RETURN 0
_SETUP_VIDEO    ENDP
;*************************************************************;
;       _FPTIME                                               ;
;       TIME EXECUTION OF FLOATING POINT DIVIDE               ;
;*************************************************************;
        EVEN
        DB      ?
                PUBLIC  _FPTIME
_FPTIME         PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, 0                   ; CLEAR DI
        MOV     AX, [BP+4]              ; GET COUNT ARGUMENT
        ADD     AX, 99                  ; ROUND UP
        MOV     CX, 100                 ; DIVIDE BY 100 =
        DIV     CL                      ;  NUMBER OF INSTRUCTIONS
        MOV     CL, AL                  ;  PER PASS
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        FNINIT                          ; INIT FP
        FLD1                            ; DIVIDE 1.0
        FLD1                            ; BY 1.0
        CLI                             ; STOP INTERRUPTS
        OUT     PPI_PORT, AL            ; ENABLE TIMER
FL:     REPT    100                     ; DO 100 DIVIDES
        FDIV    ST(1), ST               ;
        ENDM                            ; END MACRO
        DEC     CX                      ; COUNT THIS PASS
        JZ      FD                      ; JUMP IF COMPLETE
        JMP     FL                      ; LOOP BACK IF NOT DONE
```

```
FD:     MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_FPTIME         ENDP
;*************************************************************;
;       _WEMPTIME                                             ;
;       TIME EXECUTION OF PUSHA INSTRUCTION                   ;
;*************************************************************;
        EVEN
        DB      ?, ?, ?
                PUBLIC  _WEMPTIME
_WEMPTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        PUSH    DI                      ; SAVE DI
        MOV     BP, SP                  ;
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     AX, [BP+6]              ; GET COUNT ARGUMENT
        CWD                             ; MAKE DOUBLE WORD
        MOV     CX, 200                 ;
        DIV     CX                      ; DIVIDE BY MOVS/LOOP
        MOV     CX, AX                  ;
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        MOV     DX, SS                  ; SAVE STACK SEGMENT
        MOV     SS, EMMBASE             ; PUT STACK IN EMM
        MOV     SP, 400                 ; SET SP FOR PUSHES
        MOV     DI, SP                  ; SAVE THIS NUMBER
        OUT     PPI_PORT, AL            ; ENABLE TIMER
EPL:    REPT    25                      ; PUSH THE REGISTERS
        DB      60H                     ;
        ENDM                            ;
        MOV     SP, DI                  ; PUT THE STACK BACK
        LOOP    EPL                     ; LOOP UNTIL DONE
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        MOV     SS, DX                  ; RESTORE ORIGINAL STACK
        MOV     SP, BP                  ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_WEMPTIME       ENDP
;*************************************************************;
;       _DEMPTIME                                             ;
;       TIME EXECUTION OF PUSHA INSTRUCTION                   ;
;*************************************************************;
        EVEN
        DB      ?
                PUBLIC  _DEMPTIME
_DEMPTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        PUSH    DI                      ; SAVE DI
        MOV     BP, SP                  ;
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     AX, [BP+6]              ; GET COUNT ARGUMENT
        CWD                             ; MAKE DOUBLE WORD
        MOV     CX, 200                 ;
        DIV     CX                      ; DIVIDE BY MOVS/LOOP
        MOV     CX, AX                  ;
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        MOV     DX, SS                  ; SAVE STACK SEGMENT
        MOV     SS, EMMBASE             ; PUT STACK IN EMM
        MOV     SP, 800                 ; SET SP FOR PUSHES
        MOV     DI, SP                  ; SAVE THIS NUMBER
        OUT     PPI_PORT, AL            ; ENABLE TIMER
EDL:    REPT    25                      ; PUSH THE BIG REGISTERS
        DB      66H, 60H                ;
        ENDM                            ; END MACRO
        MOV     SP, DI                  ; PUT THE STACK BACK
        LOOP    EDL                     ; LOOP UNTIL DONE
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        MOV     SS, DX                  ; RESTORE ORIGINAL STACK
        MOV     SP, BP                  ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_DEMPTIME       ENDP
;*************************************************************;
;       _BEMMTIME                                             ;
;       TIME EXECUTION OF REP MOVSB INSTRUCTION               ;
;*************************************************************;
                PUBLIC  _BEMMTIME
_BEMMTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, EMMBASE             ; SET UP EMM BASE ADDRESS
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        XOR     DI, DI                  ; ES:DI -> TEST SEGMENT
        XOR     SI, SI                  ; DS:SI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSB                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_BEMMTIME       ENDP
;*************************************************************;
;       _WEMMTIME                                             ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION               ;
;*************************************************************;
                PUBLIC  _WEMMTIME
_WEMMTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        MOV     DI, EMMBASE             ; SET UP EMM BASE ADDRESS
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        XOR     DI, DI                  ; ES:DI -> TEST SEGMENT
        XOR     SI, SI                  ; DS:SI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
```

Version 3.0

New Features

Windows, Data Entry, Help Management, Menus, Text Editing, plus ...

SOURCE CODE

FREE

Vitamin C

It's good for your system!

The Vitamin C Difference

With **Vitamin C,** your applications come alive with windows that explode into view! Data entry windows and menus become a snap, and context sensitive pop-up help messages are nearly automatic.

With **VCScreen,** you'll save time by interactively painting windows and forms so what you see is what you get! Then, one button generates C source code ready to plug into your program and link with Vitamin C.

Easy enough for the beginner. Versatile enough for the professional. Vitamin C's **open-ended design** is full of "hooks" so you can intercept and "plug-in" special handlers to customize or add features to most routines.

Of course, Vitamin C **includes all source code FREE,** with no hidden charges. *It always has.* That means you'll have everything you need to adapt to special needs without spending hundreds of dollars more.

Windows

Create as many windows as you like with one easy function. Vitamin C automatically takes care of complicated tasks like saving and restoring the area under a window.

Options include titles, borders, colors, pop-up, pull-down, zoom-in, 4-way scrolling, scroll bars, sizes up to 32k, text file display & editing, cursor display, and more.

Unique built-in feature lets users move and resize windows during run-time via a definable key.

Access the current window by default or a specific window any time, even if it's hidden or invisible. Save and load windows on disk for more versatility!

Data Entry

Flexible dBase-like data entry and display routines feature protected, invisible, required, and scrolling fields. Picture clause formatting, full color/attribute control, selection sets, single field and full screen input, and unlimited data validation via standard and user definable routines. That means you aren't locked into one way of doing things.

Vitamin C even provides true right-to-left input of numeric fields with dynamic display of separators & currency symbols.

High Level Functions

Use our intergrated help management, multi-level menus, and text file routines, or build your own handlers using Vitamin C's basic windowing and data entry routines.

Standard help handler provides context sensitive pop-up help messages any time the program awaits key strokes. The help text file is stored on disk and indexed for quick access. So easy to use that a single function initializes & services requests by opening a window, locating, formatting, displaying, and paging through the message.

Multi-level "MacIntosh" & "Lotus" style menus make user interfaces and front ends a snap. Menus can call other menus, functions, even data entry screens, quickly and easily.

Text editor windows can be opened for pop-up note pads, memo fields, or general purpose editing. Features include insert, delete, word wrap, and paragraph formatting.

VCScreen

Screen Painter/Code Generator

Just as Vitamin C's reusable functions speed your programming, VCSreen makes it even *faster and easier* by automatically generating C source code for your data entry screens!

With VCScreen's interactive screen editor, you actually draw your forms. You can define input, output and constant fields, headings, boxes, lines and even a window for the form to run in.

What you see is what you get. If you don't like the position of an object, just "pick it up" with the cursor and move it! Changing colors, attributes, copying, and deleting is just as easy.

VCScreen generates readable C source code. It declares variables with names you provide and can even generate structures.

With VCScreen choosing the right functions, parameters and sequences, and Vitamin C supplying the functions to choose from, you can stop worrying about semi-colons, matching braces, and calling conventions and concentrate on creating your application!

30 Day Money Back Guarantee

Better than a brochure. More than a demo disk. If you're not satisfied, simply return the package within 30 days and receive a full refund of the purchase price.

Vitamin C $225.00

Includes ready to use libraries, tutorial, reference manual, demo, sample, and example programs, and quick reference card. For IBM PC and compatibles. Specify Microsoft, Lattice, Computer Innovations, Aztec, Mark Williams, Wizard, DeSmet, or Datalight C compiler AND compiler version number when ordering.

Vitamin C Source ... FREE*

**Free with purchase of Vitamin C*

VCScreen $99.95

Requires Vitamin C and IBM PC/XT/AT or true compatible.

ALL ORDERS:

SHIPPING: $3 ground, $6 2-day air, $20 overnight, $30 overseas. Visa and Master Card accepted. All funds must be U.S. dollars drawn on a U.S. Bank. Texas residents add 7¼% sales tax.

For Orders or More Information, Call ...

(214) 245-6090

creative PROGRAMMING

Creative Programming Consultants, Inc.
Box 112097 Carrollton, Texas 75011

```
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_WEMMTIME       ENDP
;*************************************************************;
;       _DEMMTIME                                             ;
;       TIME EXECUTION OF REP MOVSW INSTRUCTION               ;
;*************************************************************;
                PUBLIC  _DEMMTIME
_DEMMTIME       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSH    DS                      ; SAVE DS
        PUSH    ES                      ; SAVE ES
        PUSH    SI                      ; SAVE SI
        PUSH    DI                      ; SAVE DI
        CALL    SETUP_TIMER             ; SET UP TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     DI, DI                  ; CLEAR EDI
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     SI, SI                  ; CLEAR ESI
        DB      066H                    ; 32 BIT OPERANDS:
        XOR     CX, CX                  ; CLEAR ECX
        MOV     DI, EMMBASE             ; SET UP EMM BASE ADDRESS
        MOV     ES, DI                  ;
        MOV     DS, DI                  ;
        XOR     DI, DI                  ; ES:DI -> TEST SEGMENT
        XOR     SI, SI                  ; DS:SI -> TEST SEGMENT
        MOV     CX, [BP+4]              ; GET COUNT ARGUMENT
        IN      AL, PPI_PORT            ; GET CURRENT CONTROL
        MOV     BL, AL                  ; SAVE IN BL
        OR      AX, 1                   ; SET TIMER ENABLE BIT
        CLI                             ; STOP INTERRUPTS
        CLD                             ; SET FORWARD DIRECTION
        OUT     PPI_PORT, AL            ; ENABLE TIMER
        DB      066H                    ; 32 BIT OPERANDS:
        REP MOVSW                       ; RUN TEST
        MOV     AL, BL                  ; RESTORE CONTROL VALUE
        OUT     PPI_PORT, AL            ;
        STI                             ; START INTERRUPTS
        CALL    GET_TIMER               ; OBTAIN FINAL COUNT
        POP     DI                      ; RESTORE DI
        POP     SI                      ; RESTORE SI
        POP     ES                      ; RESTORE ES
        POP     DS                      ; RESTORE DS
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_DEMMTIME       ENDP
;*************************************************************;
;       _SETUP_EMM                                            ;
;       SET UP EXPANDED MEMORY AND RETURN THE BASE            ;
;*************************************************************;
                PUBLIC  _SETUP_EMM
_SETUP_EMM      PROC    NEAR
        PUSH    BP                      ; SAVE REGISTERS
        PUSH    ES                      ;
        PUSH    SI                      ;
        PUSH    DI                      ;
        MOV     AH, 35H                 ; GET EMM INTERRUPT
        MOV     AL, 67H                 ; VECTOR
        INT     21H                     ;
        MOV     AX, ES                  ; CHECK FOR VALID BASE
        MOV     BX, CS                  ; MUST BE BELOW CS
        CMP     AX, BX                  ;
        JA      SENO                    ; JUMP IF NOT GOOD
        MOV     DI, 000AH               ; OFFSET OF DRIVER NAME
        LEA     SI, EMM_NAME            ; COMPARE STRING
        MOV     CX, 8                   ; LENGTH OF STRING
        CLD                             ;
        REPE    CMPSB                   ; COMPARE THE NAME
        JNE     SENO                    ; JUMP IF NO GOOD
SE1:    MOV     AH, 40H                 ; FUNCTION 1:
        INT     67H                     ; GET MANAGER STATUS
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE1                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENO                    ; JUMP ON ERROR
SE2:    MOV     AH, 41H                 ; FUNCTION 2:
        INT     67H                     ; GET PAGE FRAME BASE
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE2                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENO                    ; JUMP ON ERROR
        MOV     EMMBASE, BX             ; SAVE THE BASE
SE3:    MOV     AH, 42H                 ; FUNCTION 3:
        INT     67H                     ; GET NUMBER OF PAGES
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE3                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENO                    ; JUMP ON ERROR
        OR      BX, BX                  ; CHECK UNALLOCATED PAGES
        JZ      SENO                    ; JUMP IF NONE AVAILABLE
SE4:    MOV     AH, 43H                 ; FUNCTION 4:
        MOV     BX, 1                   ; ALLOCATE ONE PAGE
        INT     67H                     ;
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE4                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENO                    ; JUMP ON ERROR
        MOV     PID, DX                 ; SAVE THE PROCESS ID
SE5:    MOV     AH, 44H                 ; FUNCTION 5:
        XOR     BX, BX                  ; MAP THE PAGE TO
        XOR     AL, AL                  ;  FRAME BASE
        INT     67H                     ;
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE5                     ; TRY AGAIN IF BUSY
        OR      AH, AH                  ; CHECK FOR ERROR
        JNZ     SENC                    ; JUMP ON ERROR
        XOR     AX, AX                  ;
        POP     DI                      ; RESTORE REGISTERS
        POP     SI                      ;
        POP     ES                      ;
        POP     BP                      ;
        RET                             ; RETURN 0
SENC:   MOV     AH, 45H                 ; FUNCTION 6:
        INT     67H                     ; CLOSE EMM
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SENC                    ; TRY AGAIN IF BUSY
SENO:   MOV     AX, OFFFFH              ;
        POP     DI                      ; RESTORE REGISTERS
        POP     SI                      ;
        POP     ES                      ;
        POP     BP                      ;
        RET                             ; RETURN -1
_SETUP_EMM      ENDP
;*************************************************************;
;       _FINISH_EMM                                           ;
;       CLOSE THE EMM DEVICE, RELEASE THE PAGE                ;
;*************************************************************;
                PUBLIC  _FINISH_EMM
_FINISH_EMM     PROC    NEAR
        PUSH    BP                      ; SAVE REGISTERS
        PUSH    ES                      ;
        PUSH    SI                      ;
        PUSH    DI                      ;
SE6:    MOV     AH, 45H                 ; FUNCTION 6:
        MOV     DX, PID                 ; CLOSE EMM
        INT     67H                     ;
        CMP     AH, 82H                 ; CHECK FOR BUSY
        JE      SE6                     ; TRY AGAIN IF BUSY
        POP     DI                      ; RESTORE REGISTERS
        POP     SI                      ;
        POP     ES                      ;
        POP     BP                      ;
        RET                             ; RETURN
_FINISH_EMM     ENDP
;*************************************************************;
;       SETUP_TIMER                                           ;
;       SET UP THE TIMER FOR MAXIMUM COUNT, TO TIME A RUN     ;
;*************************************************************;
SETUP_TIMER     PROC    NEAR
        PUSH    AX                      ; SAVE AX
        IN      AL, PPI_PORT            ; STOP THE TIMER
        AND     AL, OFCH                ;
        OUT     PPI_PORT, AL            ;
        MOV     AL, 0B4H                ; INITIALIZE THE TIMER
        OUT     TIMER_CTRL, AL          ;
        MOV     AL, 0                   ; CLEAR THE COUNT
        OUT     TIMER2_PORT, AL         ;
        NOP                             ;
```

```
        OUT     TIMER2_PORT, AL         ;
        POP     AX                      ; RESTORE AX
        RET                             ; RETURN
SETUP_TIMER     ENDP
;*************************************************************;
;       GET_TIMER                                             ;
;       TAKE THE COUNT FROM THE TIMER                         ;
;*************************************************************;
GET_TIMER       PROC    NEAR
        PUSH    BX                      ; SAVE REGISTERS
        IN      AL, TIMER2_PORT         ; GET LOW BYTE OF TIME
        MOV     AH, AL                  ;
        IN      AL, TIMER2_PORT         ; GET HIGH BYTE
        XCHG    AL, AH                  ; TIME IN AX
        NEG     AX                      ; CORRECT FOR COUNT-DOWN
        POP     BX                      ; RESTORE REGISTERS
        RET                             ; RETURN
GET_TIMER       ENDP
;*************************************************************;
;       _NDP_PRESENT                                          ;
;       CHECK IF 80287 IS PRESENT                             ;
;*************************************************************;
                PUBLIC  _NDP_PRESENT
_NDP_PRESENT    PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        INT     11H                     ; BIOS EQUIP CHECK
        TEST    AL,02H                  ; IS 80287 BIT SET?
        JZ      NO                      ; NO MEANS NO 80287
        MOV     AX,01h                  ; RETURN TRUE
        JMP     NDPEXIT                 ; ALL DONE
NO:     XOR     AX,AX                   ; SET AX TO FALSE
NDPEXIT:MOV     SP, BP                  ; RESTORE SP
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_NDP_PRESENT    ENDP
;*************************************************************;
;       _CPU_TYPE                                             ;
;       CHECK IF CPU IS 8088/8086, 80188/80186, 80286, 80386  ;
;*************************************************************;
                PUBLIC  _CPU_TYPE
_CPU_TYPE       PROC    NEAR
        PUSH    BP                      ; SAVE FRAME
        MOV     BP, SP                  ;
        PUSHF                           ;
        XOR     AX,AX                   ; ZERO AX
        PUSH    AX                      ;
        POPF                            ; TRY TO PUT 0 INTO FLAGS
        PUSHF                           ;
        POP     AX                      ; SEE WHAT WENT IN FLAGS
        AND     AX,0F000H               ; MASK OFF HIGH FLAG BITS
        CMP     AX,0F000H               ; WAS HIGH NIBBLE ONES
        JE      _8X                     ; IS 8086 OR 8088
        PUSH    SP                      ; SEE IF SP IS UPDATED
        POP     BX                      ; BEFORE OR AFTER IT IS
        CMP     BX,SP                   ; PUSHED
        JNE     _18X
        MOV     AX,0F000H               ; TRY TO SET HIGH BITS
        PUSH    AX                      ;
        POPF                            ; IN THE FLAGS
        PUSHF                           ;
        POP     AX                      ; LOOK AT ACTUAL FLAGS
        AND     AX,0F000H               ; ANY HIGH BITS SET?
        JE      _286                    ; IS 80286
_386:   MOV     AX,03                   ; IS AN 80386
        JMP     CTEXIT                  ;
_286:   MOV     AX,02                   ; IS AN 80286
        JMP     CTEXIT                  ;
_18X:   MOV     AX,01                   ; IS AN 80188/80186
        JMP     CTEXIT                  ;
_8X:    MOV     AX,00                   ; IS AN 8088/8086
CTEXIT: POPF                            ; RESTORE ORIGINAL FLAGS
        MOV     SP, BP                  ; RESTORE SP
        POP     BP                      ; RESTORE BP
        RET                             ; RETURN
_CPU_TYPE       ENDP
_TEXT           ENDS
                END
```


C Cross Compiler 68000/08/10/20

Now Generates: Position Independent Code

Features:

- Full, Standard C
- Easy to Use Compiler Options
- Complete User Documentation
- Global Code Optimization
- Optional Register Allocation Via Coloring
- ROMable and Reentrant Code
- Comprehensive Royalty Free Run-time Library
- Floating Point Library Routines
- Intermix MCC68K C with ASM68K Assembly Language or Microtec PAS68K Pascal
- Optional Assembly Language Listing Intermixed with MCC68K C Source Line Number
- Symbolic Debug Capability

The Microtec MCC68K C Cross Compiler is a complete implementation of the 'C' programming language as defined in The C Programming Language by Kernighan and Ritchie with extensions.

MCC68K emits highly optimized assembly language code for the Microtec ASM68K Motorola compatible assembler.

The Microtec MCC68K package includes the compiler, relocatable macro assembler, linking loader, run-time library, and comprehensive user's guide.

Host computers include: DEC VAX, DG MV-Series, Apollo, IBM PC and PC-compatibles..

We're **Functional** and **Fast** and **Serious** about our products. We've been providing flexible and economical solutions for software developers since 1974.

Beginning with product concept, through development, quality assurance, and post-sales support - **Quality, Compatibility** and **Service** are the differences which set Microtec Research apart from others.

If you're a serious software developer, shopping for software development tools, call or write today for more information:

800-551-5554,

In CA call (408) 733-2919.

3930 Freedom Circle, Suite 101, Santa Clara, CA 95054
Mailing Address: P.O. Box 60337, Sunnyvale, CA 94088

MICROTEC® RESEARCH

Creating With
Pictures
Just Got Easier!

Professional Image Board



The new Professional Image Board is a PC board which allows an ordinary home video camera (color or black-and-white) to be plugged into an IBM personal computer or IBM compatible. Now, live, fast action scenes can be instantly captured in full color and frozen. The frozen pictures can be computer enhanced feature by feature and stored on a floppy or hard disk. The frozen pictures can also be transmitted to any remote computer in the world via modem. The Professional Image Board also allows you to perform cut and paste operations with most popular software.

SPECIFICATIONS

- Works with IBM PC/XT/AT or compatible
- Capture one picture in 1/60 second from standard NTSC signal source (Regular video camera, VCR, etc., or RGB camera)
- Shows picture on analog monitor or TV
- Resolution 512 x 256 capable of 32K different colors
- Adding text on the captured picture
- Mixing external image with internal image in live mode
- Image captured by PIB can be displayed on EGA monitor
- Adapter available for European standard TV



Unretouched frozen video image in full color. Captured with ATronics' Professional Image Board.

ATI **ATronics International Inc.**
We Deliver Advanced Technology

1830 McCandless Dr. Milpitas CA 95035
(408) 943-6629

CIRCLE NO. 238 ON READER SERVICE CARD

Multilevel Debugger

Window-oriented CodeView heralds a new generation of products with full debugging capabilities at both the source code and assembly levels.

MARK S. ACKERMAN

Traditionally, PC software developers have not had many choices for debugging their high-level language programs. Most debuggers work in assembly language code; the programmer must be familiar with intricate details of how the compiler generates code in order to debug programs successfully. Commands are given to the debugger through a command line using cryptic incantations. In an age of mice, pull-down menus, and windows, most debuggers are throwbacks to the days of DOS 1.0.

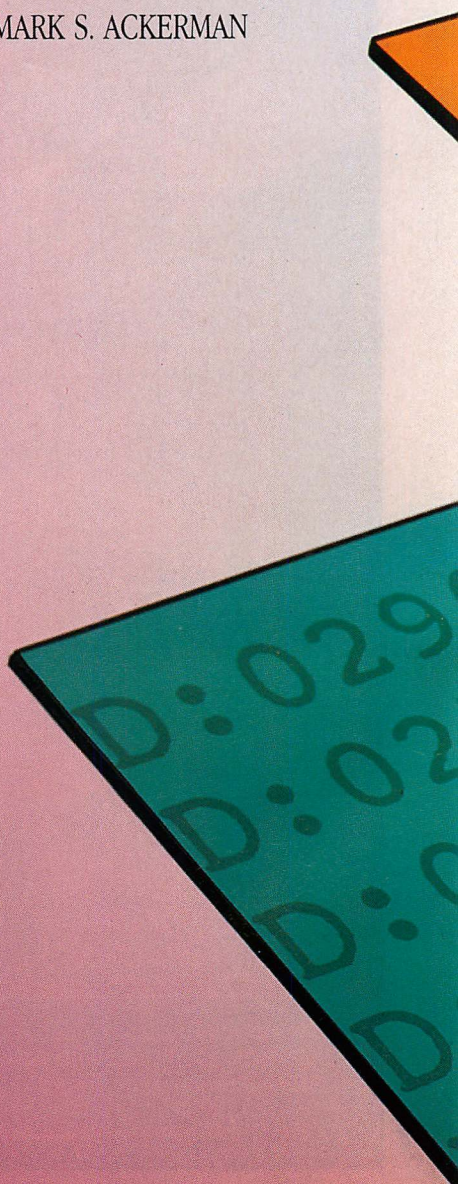
CodeView is Microsoft Corporation's attempt to advance the state of the art in debuggers. CodeView provides debugging at both the source and assembly language levels. For users who are familiar with assembly-level assistance, CodeView is compatible with both Microsoft's DEBUG and SYMDEB line-oriented debuggers. For high-level language programmers, CodeView allows almost complete debugging at the source-code level.

Assembly-level debuggers such as DEBUG always work at the level of machine instructions (either hexadecimal or assembly language mnemonics) and memory locations. Although appropriate for programs written in assembly language, DEBUG is not well suited to higher-level languages. The original structure of the program is obscured by compilation; at best, the compiler may produce a listing that shows the assembly language code generated from the source. In this case, the programmer must understand not only the source-level program but also the resulting assembly language code that is produced by the compiler.

SYMDEB, currently provided by Microsoft with its Macro Assembler version 4.0, extends the capabilities of DEBUG by adding limited support for debugging at the source-code level. The programmer can step through one source-code line at a time. Symbols for function names and global data can be used instead of absolute memory addresses. However, the debugger does not know the size or type of data designated by the symbol; the programmer must provide that information.

CodeView also allows debugging at the source-code level, but instead of examining bytes or words in memory, the programmer can examine variables as their program-defined types: integers, floats, strings, and structure members. Instead of stopping on an assembly language instruction, the programmer can stop on source lines. Both global and local variables may be examined, CodeView knows the size and structure of variables in the program.

Often, an assembly-level debugger is not necessary for application programming. Even for systems programming done in C, source-level debugging



PHOTOGRAPH • WALTER LARRIMORE/BLAKESLEE-LANE

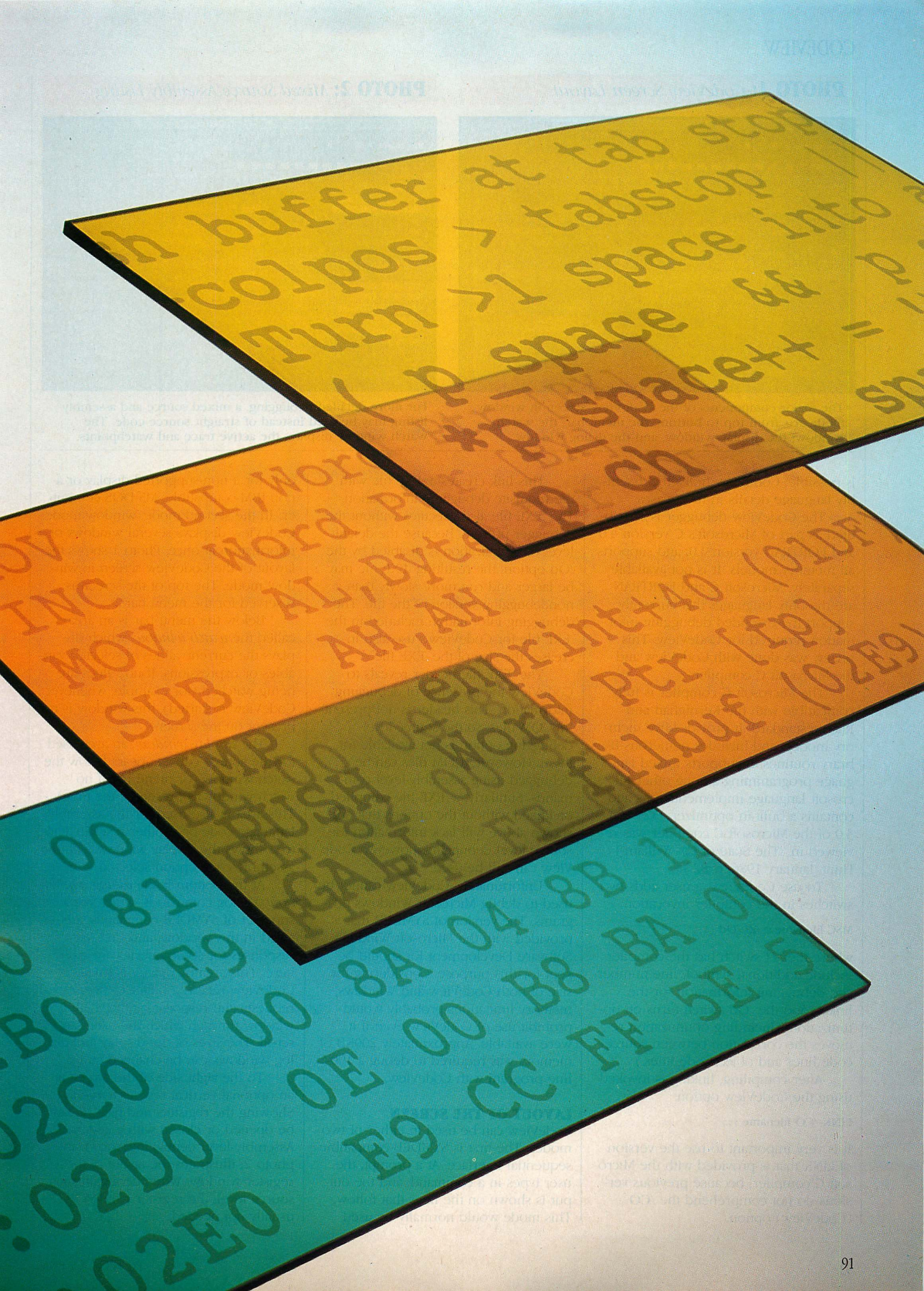
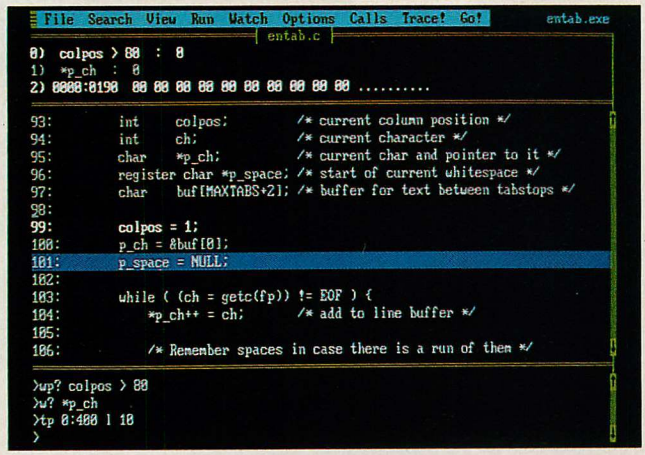
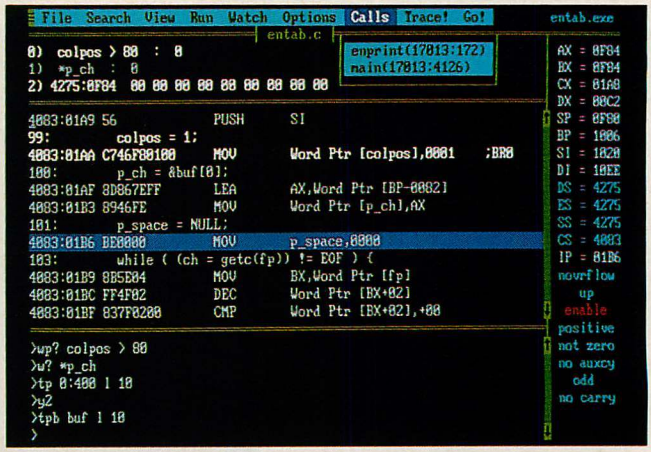


PHOTO 1: *CodeView Screen Layout*



The major components of the CodeView screen in window mode are, from top to bottom, the menu bar, the watch window, the display window, and the dialog window.

PHOTO 2: *Mixed Source/Assembly Listing*



For more detailed debugging, a mixed source and assembly listing may be used instead of straight source code. The watch window displays the active trace and watchpoints.

is possible. CodeView hides the assembly language details in most cases.

The CodeView debugger is distributed as part of Microsoft's C version 4.0 and FORTRAN version 4.0, and supports all memory models. It is not available separately. Microsoft C and FORTRAN are the only languages that currently support the extended debugging information required for CodeView. This review was done with CodeView and the Microsoft C compiler.

The Microsoft C compiler is fully compatible with the Kernighan and Ritchie standard. It provides five memory models and includes numerous library routines. It supports mixed language programming among other Microsoft language implementations and contains a built-in optimizer. (Version 3.0 of the Microsoft C compiler was reviewed in "The State of C," William J. Hunt, January 1986, p. 82.)

To use CodeView, a user adds two switches in the compiler invocation:

```
MSC filename.c /Zi /Od ; ; ;
```

where the **/Zi** switch has the compiler include additional symbol, line number, and debugging information for the linker, and the **/Od** switch turns off optimization. (Including optimization removes the correlation between source-code lines and object-code lines.)

After compiling, linking is invoked using the CodeView option:

```
LINK /CO filename ; ; ;
```

It is very important to use the version of LINK that is provided with the Microsoft C compiler, because previous versions do not comprehend the **/CO** (CodeView) option.

This link creates a .EXE file with the requisite debugging information. The .EXE file also executes without the debugger present. Because the default level of optimization is disabled by the **/Od** option, the resulting .EXE file may be larger and run more slowly than a nondebugging version of the file. The debugging information included in the .EXE file for CodeView's use also increases the size of the .EXE file.

The .EXE file header reveals to CodeView the presence of debugging information. This header is not present in .COM files, so the source-level debugging option is not available for them. However, .COM files can be debugged using assembly-level commands. Similarly, a .EXE file compiled or linked without the options that are shown above will lack information for source-level debugging; however, CodeView can still be used.

Unfortunately, CodeView cannot be used to debug Microsoft Windows programs. The version of SYMDEB that is provided with the Microsoft Windows Software Development Kit should be kept for this purpose. Considering the size of both CodeView and Windows, memory limitations probably would prohibit use of CodeView even if it were available. Approximately 200KB of memory was required to debug a 100-line program with CodeView.

LAYOUT OF THE SCREEN

CodeView can be used in either of two modes. The first is SYMDEB-compatible sequential interface. At a prompt, the user types in a command, and the output is shown on the lines that follow. This mode would normally be used only for a non-supported display or a non-IBM-compatible, MS-DOS computer. In the second mode, window mode, CodeView displays several windows of information at once. Photo 1 shows the layout of the CodeView screen in window mode. The top of the screen is reserved for the menu bar.

Below the menu bar is an area called the *watch window*, which displays the current values of selected variables or expressions. If no values are being watched (for example, whenever CodeView is started), this window is not present on the screen.

The *display window*, an area used for the source code, appears below the watch window (or menu bar if no watch window is present at the time). This area is also used for the assembly language code or another text file. Photo 2 shows the display window with the assembly code option selected.

At the bottom is the *dialog window*. Dialog commands, which are a superset of SYMDEB commands, are entered in this window, and CodeView's responses for some queries are given here. This area by default is small. CodeView retains a buffer of previous requests and responses in the dialog window through which the user can scroll. The sizes of the display and dialog windows can be changed easily.

To the right side of the screen is an optional vertical *register window*. Showing the registers and flags, it can be opened or closed with a keystroke. Assembly-level debugging, shown in photo 2, illustrates the layout of the register window. When debugging at source level, the register window is usually not needed.

DAN BRICKLIN'S DEMO PROGRAM HAS EVERYONE TALKING.

Dan Bricklin's Demo Program is a new concept in prototyping and demo-making. With DEMO you can conceptualize programs, describe them to others, refine their functionality and human interface, and finally, teach users the finished product. The two most common applications are to prototype a product before it is written and to produce a demonstration or tutorial about an existing product or system.

- Create, edit and view a series of slides
- Each slide is 80x25 characters
- All 256 characters supported, including special characters
- All 256 attributes (color, underline, blink, etc.)
- Each slide can be the image of what a running program looks like
- Characters graphics only, no bit-mapped graphics
- Slides are stored compacted, allowing often hundreds and hundreds of slides to fit in memory for instant access (given enough RAM)
- Complete set of editing commands, including cut, paste and move
- Line, box and special character commands built-in
- Includes background "CAPTURE" program for importing screens from running programs
- Learn-mode macro facility
- Run command switches from slide to slide under automatic control

  Options include:
  - ☐ Adjustable delay between each slide
  - ☐ Wait for key or optional timeout
  - ☐ Display different slide next depending upon keystroke
  - ☐ Produce tones
  - ☐ Switch to another set of slides in another file
- Runtime-only version of program included, along with license to make up to 50 copies (see License for details).
- Additional volume runtime licenses available.
- Requires 256k IBM PC/compatible, DOS 2.0 or later. Supports Monochrome, Color/graphics, and EGA Adaptors (text mode only).

Thousands of developers are designing better products faster and producing more effective demonstrations using Dan Bricklin's Demo Program. You can, too. Act now!

"A winner right out of the starting gate.. After you use DEMO once, you'll wonder how you got along without it."—PC Magazine, 4/29/8

PRODUCT OF THE MONTH
—PC Tech Journal, 3/86

"Everybody who writes software, either commercially or for in-house applications, should immediately order a copy. Period. No exceptions."
—Soft•letter, 4/20/86

"Its low price, superb performance, and range c applications practically guarantee that it will be widely used. Four Floppy Rating (8.0)"
—InfoWorld, 3/31/86

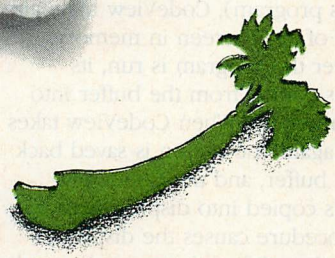
"Apparently has a hit on its hands with... a development tool for personal computer software that has won rave reviews from early users."
—Computerworld, 4/7/86

ORDER YOUR COPY TODAY

MasterCard VISA

ONLY $74.95
617-332-2240

Send check to Software Garden, Inc. Dept. TD, P.O. Box 373, Newton Highlands, MA 02161. Massachusetts residents add $3.75. Outside the U.S.A. add $15.00.



SOFTWARE GARDEN

"A gem."
—PC Week, 3/18/86

CIRCLE NO. 142 ON READER SERVICE CARD

SEEING CODEVIEW

CodeView was tested on an AT with a Hercules Graphics Card, an AT with an IBM Enhanced Graphics Adapter (EGA), and an AT&T PC6300 with an AT&T adapter compatible with the IBM Color Graphics Adapter (CGA) and monochrome display. Both AT configurations were run under DOS 3.1, and the PC6300 with AT&T's MS-DOS 2.11.

On the ATs, CodeView recognized that DOS was being used and automatically entered the window mode when it was started. On the PC6300, the /W (window) and /B (black-and-white display) options had to be used to start CodeView in window mode.

Screen output is a difficult area for debuggers such as CodeView. When the program being debugged is not running, CodeView uses the entire screen to display debugging information. Whenever the program being debugged is started, a separate screen environment is provided for it. When the debugger resumes control (such as after a breakpoint), the program's screen is saved so that CodeView can reclaim the screen for debugging use.

If all of the application's screen output were done through the BIOS, CodeView could detect a BIOS video call (INT 10H) and restore the program's screen only when needed. In reality, many programs write directly to display adapter memory, avoiding the BIOS. CodeView could check every instruction for writes to display memory, but this would significantly slow program execution.

Instead, CodeView provides a good selection of compromises for handling screen output. On a monochrome adapter or an EGA/CGA program that changes display modes (for example, a graphics program), CodeView maintains a buffer of each screen in memory. Whenever the program is run, its screen is copied from the buffer into display memory. When CodeView takes control again, the screen is saved back into the buffer, and the CodeView screen is copied into display memory. This procedure causes the display to jump back and forth between the application and CodeView screen in a disconcerting manner.

With the CGA and EGA, CodeView uses two display pages in the video adapter memory: one for itself and the other for the program output. It flips between the two pages whenever the program is started, producing a much smoother transition between screens. The EGA also can be used in 43-line mode. The greater number of lines on the debugging screen in 43-line mode is helpful, but the annoying jumping between the CodeView screen and a 24-by-80 program screen becomes even more pronounced.

If a program is being debugged in a section of code where screen output is not being performed, the programmer can have CodeView disable the annoying screen-swapping that occurs during program execution. If it is carefully used, disabling screen-swapping can make CodeView much easier to work with. However, if the program does write directly to the screen, the output either overwrites part of CodeView's display or it disappears as CodeView updates its display.

The best option for debugging is one that is not mentioned in the Microsoft manual: using two displays. A CodeView command line switch, /2, places program output on the default display (selected by the DOS MODE command), and CodeView's output appears on the second display. Information about this feature is contained in the README.DOC file supplied on disk, rather than in the documentation.

Most commands can be executed by selecting from the pull-down menu bar, pressing function keys, or by typing the commands.

USING CODEVIEW

CodeView is started with the command:

CV options executablefile arguments

Options are provided to select window mode, sequential mode, black-and-white display, screen flipping and/or swapping, or IBM EGA 43-line mode. One useful option allows a sequence of commands to be executed when CodeView is started.

The name of the executable file to be debugged (including the .COM or .EXE extension) is given after any options. If the program expects any command-line arguments, they can be given after the executable file name.

Typically, the executable file is debugged in the same directory in which the source files reside. If CodeView cannot find the source files used to create the executable file, it prompts for the file's location. If a source file is not available, only assembly language debugging can be performed.

Every user should be able to find a comfortable interface with CodeView. Most commands can be executed by selecting from the pull-down menu bar, pressing function keys, or typing commands in the dialog window. Menu selections can be made with a Microsoft-compatible mouse or by using Alt-key combinations.

A mouse is used just as it is in Microsoft Windows. Items in the menu are selected by pointing to the main menu item, pressing a mouse button, and moving the selection bar to the desired item. Releasing the mouse button causes the selection to be made. The display and dialog windows have scroll bars that can be used to scroll through the source/assembly language listing or through previously entered dialog commands. The display/dialog separator line can be moved by the mouse to adjust the relative sizes of the two windows. The mouse also can be used to change any flag in the register window.

Microsoft has provided a new MOUSE.SYS driver with CodeView that can save and restore the mouse cursor, so that both CodeView and the program being debugged can use the mouse. Unfortunately, CodeView may cause problems for mice that do not use the Microsoft driver. Logitech Inc.'s latest Logimouse software has been made compatible. At the time of this review, however, Mouse Systems Corporation did not have PC Mouse software that worked with CodeView.

For keyboard use, the menu items have been named so that pressing the Alt key plus the first letter in the name will open the menu. Pressing the Alt key plus the first letter of the item in the opened menu will select that item. Commonly used commands are duplicated in the function keys, which allow actions such as set/clear breakpoints, single-step, and "run until this instruction is reached" to be performed with a single keystroke.

The familiar SYMDEB command-driven interface is available through the dialog window. The major changes and additions made to the SYMDEB commands are listed below:

- The E (enter) command, when typed with no arguments, executes the program in slow motion, highlighting each instruction as it is executed. If watchpoints and/or registers are being displayed, they are also updated.
- Commands have been added to manipulate tracepoints and watchpoints.

- The 7 command can be used to display detailed information about the 8087 or 80287 numeric coprocessor registers and internal state.
- In many of the commands that require values, a C-style expression often can be used. The expression can include program variables, register names, and constants.
- In keeping with the high-level debugging nature of CodeView, the default base for constants is 10. For hexadecimal constants, the C convention of preceding the number with **0x** should be used. Alternatively, the default base for constants can be changed to 16 with the **N** (numeric radix) command.
- The **W** (write) command of DEBUG and SYMDEB is not supported. DEBUG or SYMDEB must be used for modifying executable files or patching absolute disk addresses.
- The **S** (Search) command is not supported by CodeView.

EXECUTING CODE

Program execution begins with a function key, menu, or dialog command and stops whenever a breakpoint, watchpoint, or tracepoint is encountered. The user can specify that the execution be done in slow motion or that it stop on the current cursor line (a temporary breakpoint). The current line of code being executed is identified by reverse video highlighting.

The Trace command lets the user step through the program one or more lines at a time. It steps into C functions for which there is source code, if the display window contains C code. If CodeView is at the assembly level, the Trace command moves into any function or interrupt except a DOS interrupt 21. CodeView moves through a macro's expansion only at the assembly level, and the source line expansion is not available. After each step, all watch expressions are updated.

The Program Step command, does not enter C or assembly language procedures. While it is similar to the Trace command, it steps over the called procedure to the next source line in the current procedure.

Once entered, a procedure must be stepped through or a breakpoint must be set in the calling procedure. This can be done by making use of the Call Stack command.

If the program enters an endless loop or if a problem is encountered during execution, control can be regained through Ctrl-C or Ctrl-Break. If the program has reset these keys, CodeView can be stopped with the AT's SysReq key. CodeView does not abort; it gracefully returns control to the user.

At any point, the program can be restarted, which starts execution from the beginning of the program, as if the debugger has just been entered. Breakpoint and watch expression definitions still are retained.

Often while debugging, the user needs to know the calling sequence and parameters, as well as the relation among procedures, not just the procedures themselves. The calls menu or Stack Trace command shows the calling sequence to the present execution location. Photo 2 shows the CodeView screen with the calls menu pulled down. The menu is dynamic, showing the current function at the top and the calling procedures below. **Main( )** is always the last function shown in the calls menu. The values of the parameters are also given in the menu.

Unlike the other menus, the calls menu cannot be used by entering the combination of Alt key and the first letter of the menu item. The user must move the selection bar up or down the menu and then press the Enter key. This moves the cursor to the source code at the calling point (or the calling instruction if called from assembly language). The calling parameter list can then be examined. If the uppermost menu item is selected, CodeView returns to the current position.

SEARCHING CODE

Examining code is a straightforward procedure in CodeView. The user is able to search through the code with the search menu or dialog commands or by using the cursor keys.

The view menu or the F3 key allows the user to move from C to assembly language and back. The assembly language code is mixed code; the C source is contained in the assembler, where possible, as comment lines. Photo 2 shows the mixed source/assembly format. The C source comments can be turned off completely to get straight assembly language without symbolic references. When source code is not available, the user sees only assembly language instructions in the standard Unassemble format from DEBUG. This is true for all .COM files.

The Search function lets the user find a variable, label, or expression in the code. After selecting the Find or Label menu item, a dialog box appears, and then the search string is specified by entering a regular expression.

Regular expressions are standard in the UNIX environment; they form a comprehensive but enigmatic search language. Regular expressions can include wild cards and special search characters. An example might be **^buffer**, which searches for the word *buffer* at the beginning of a line. In the simplest case, a normal text string can be entered. To look for the occurrences of the C library function **fopen( )**, the user can type **fopen**. However, to look for **buffer[i]**, the user would need to know to enter **buffer\[i]**. Similarly, the * does not work quite the same way as at the DOS level. Regular expressions are very strong in their searching and matching capabilities, and their inclusion is quite useful. The CodeView manual includes an excellent appendix on regular expressions.

Two types of searching can be performed with CodeView. The first is a text search that uses a regular expression to find any string in the source file. Through menu selections, CodeView can perform a text search both forward and backward in the source file. With a dialog command, only forward searches are possible. Oddly, the search-backward menu selection does not allow a search string to be entered; it uses the string entered for the last forward-search command. Because searching wraps around to the top of the file (instead of giving a "not found" message), a search-forward command may find an occurrence at an earlier line in the file. This mirrors the behavior of searches in UNIX-style editors, but may take some users by surprise.

For example, a search for the string **strcpy** would find the first mention of *strcpy* in the source file starting from the cursor position, whether it is a function call, a comment, or a declaration. Subsequent searches will find later occurrences of *strcpy*, until the search wraps around to the first occurrence.

The second form of search is a label search, which finds the definition of a function or label. The name given for the label must be complete and cannot be a regular expression. If a label has been defined in C, CodeView finds that location, changing source files if necessary. If, however, the label is defined in assembly language or in executable code, then CodeView changes to assembly language to display the code. For example, a search for **strcpy** as a label would move to the location of the C library routine, and the debugger would change to assembly level.

Modifying code can be done only at the assembly level. The Assemble Dialog command assembles 8086-family mnemonics. Some programmers find

this command invaluable; they use it to jump to a patch area where they can enter temporary bug-fixing code. Instruction mnemonics for the 8087 and 80287, however, are not supported by the in-line assembler.

DATA EXAMINATION

CodeView provides programmers with a familiar environment in which to examine data using several methods. The first is the old DEBUG command for dumping memory. Entering dialog commands such as

```
D DS:0x1F0 230
```

produces a hexadecimal dump of that portion of memory. CodeView follows SYMDEB in having ASCII, integer, word, real, and several other types of dumps.

Virtually all of the familiar SYMDEB commands have been included in CodeView—for example, the Display Expression dialog command. (This facility is also available through the Evaluate menu item on the view menu.) The Display Expression command displays the value of a subset of C expressions. An expression includes constants, variables, most C operators, and some type casts (structure or union type casts are not allowed). For example, the following command

```
? global_var+5
```

prints the value 15 in the dialog window if **global_var** is currently 10.

At a simple level, the Display Expression command operates as a calculator. It is also possible to determine the address of a variable using the **&** operator. In addition, the result can be formatted in a manner similar to the **printf** statement. For example, the dialog command, **? 100 , x** , asks for the value of 100 decimal in hexadecimal. The **x** asks that result be printed as a hexadecimal. All of the normal **printf** types are possible: decimal integer, hex, octal, float, exponential, string, character, and various longs. Most importantly, the value of a variable ccan be determined in C notation with the Display Expression command. The expression

```
? ptr —>member[index] , d
```

is perfectly valid. Individual address calculations are unnecessary.

Because the C assignment and increment operators are supported in expressions, display expression also can be used to modify data. This point is not well made in the manual. In fact, the manual does not refer to this useful feature in the chapter on modifying data. Although the **E** (enter) command of SYMDEB is also supported, the advantage of Display Expression for changing data is that the type and size of the data are known, whereas Enter requires the programmer to specify the type and size.

The command **? var1 = 5** sets **var1** to 5. In a similar manner,

```
? structB.var2++
```

increments the current value of **var2** in **structB**. Display Expression does not allow function calls to be executed. Because C strings are supported by library function calls, rather than directly in the language, the only way to modify a string is to use the Enter command:

```
E string_name "the rain"
```

This creates a small inconsistency in the way data are modified.

CodeView also has facilities for examining data through watch expressions. The ability to evaluate an expression or variable whenever a breakpoint or tracepoint is reached is available through the watch menu or a dialog command. This evaluation occurs automatically whenever CodeView is active.

Watch expressions are shown in the watch window directly below the menu bar, sharing the available screen space with the display window. Therefore, the more watch expressions that are used, the smaller the display window. Four or five watch expressions are a reasonable number for a 25-line screen; the EGA's 43-line mode is especially helpful when several watch expressions need to be displayed.

Limitations on the visibility of variable names in CodeView can make entering expressions somewhat awkward. Watch expressions for a procedure's local variable cannot be entered until the procedure is called. In fact, at least one line of code in the procedure must be executed before any watch expressions can be entered.

To know the value of a variable in a calling procedure, the user must prefix the variable name with the procedure name:

```
W? procedure_1.variable_A
```

If this is not done, CodeView does not know **variable_A**'s value within **procedure_1**, even in a procedure that is called by **procedure_1**. As a matter of convenience, CodeView should allow the user to set the locality for variables in the watch expression through the location of the cursor in the source. CodeView should also allow watch expressions to be set for functions that have not been reached, so that a watch expression could be set up before debugging actually starts.

BREAK, TRACE, AND WATCH

Three facilities are provided to stop execution of a program whenever user-defined conditions occur: breakpoints, tracepoints, and watchpoints. Breakpoints are also found in DEBUG and SYMDEB. CodeView allows the user to set breakpoints visually with the F9 key, a mouse button, or a dialog command. The source line is then highlighted on the screen (see photo 1).

As many as 20 breakpoints can be set simultaneously. They remain in force until they are canceled or until the programmer leaves the debugger. They are retained when the program is restarted. Breakpoints can be temporarily disabled and then reenabled.

When the program is about to execute a code statement that contains a breakpoint, CodeView stops, sets the display window to the appropriate point in the code, and updates any watch expressions. The user can then enter any CodeView commands. By appropriately placing breakpoints and watch expressions, users easily can gain snapshots of their programs.

An additional feature of breakpoints is that, through the dialog command, CodeView can stop at the *n*th iteration of a code line. The dialog command **BP .211 75** causes CodeView to stop when line 211 is executed for the 75th time. This is ideal for stopping near the end of a loop's execution. Through a dialog command, the user also can ask for the automatic execution of other commands when a breakpoint is triggered. For example,

```
BP .1034 "? i;? j=4"
```

prints out the value of *i* and resets *j* to 4 every time the breakpoint is encountered in the program.

The second form of control is the tracepoint. It stops program execution whenever a memory location (or a range of memory up to 128 bytes) is modified. The expression defining the area of memory to be traced can be either a program variable name or a set of absolute memory addresses. Variables and memory ranges being traced appear in the watch window.

Watchpoints, the third and most powerful form of control, are expressions that are evaluated after every program instruction is executed. If the expression evaluates to any value except zero, the program is stopped. Watchpoints share the watch window with tracepoints and watch expressions.

Watch expressions display only the value of a variable, whereas watchpoints can cause the program to be stopped when their expressions are not zero. Watchpoints are shown in high-intensity text in the watch window to differentiate them from the similar-looking watch expressions. The combined total number of watchpoints, tracepoints, and watch expressions cannot exceed 10.

The cost of these features is reduced execution speed of the program being debugged. Breakpoints are relatively inexpensive unless they are contained in a loop with a very high pass count. Tracepoints and watchpoints slow the program down by a factor of approximately 1,000. Depending on the program, this may be significant. Programs that deal with external events (for example, serial ports, mice, or real-time clock interrupts) may not tolerate the delay. (CodeView can take advantage of debugging hardware that reduces the overhead of tracepoints and watchpoints. Atron's MiniProbe, an example of this kind of hardware, will be reviewed in an upcoming Product Watch.)

POWERFUL FUNCTIONS

Among CodeView's features are commands to set tabs, reset register values, copy CodeView output to both file and screen, and pause. The shell menu item allows the execution of DOS commands and a second COMMAND.COM.

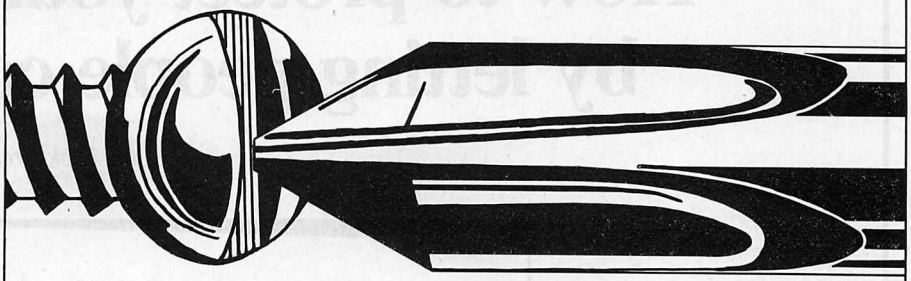
One example of the power of the I/O redirection and pausing commands is the CodeView demonstration disk. The entire demonstration is orchestrated by using *script* files read in by CodeView. A script file is useful for quickly executing a program to the point where it was failing when it requires intermediate breakpoints and patches. All of the breakpoints and patches could be done in the script file.

CodeView's Load command loads ASCII files that contain no special formatting characters. This is useful for viewing C header files that have been included by the source program.

The on-line help screens are disappointing. Relatively little information is present; in general the screens serve as syntax reminders. For a complex product, CodeView offers very few screens.

CodeView's written documentation, with the exception of the Display Expression command, is good. The manual, which currently takes up one-half of one of the three Microsoft C manuals, includes sections on using the compiler and linker with CodeView, CodeView options, screen layout, and menus. In addition, each command has

ISN'T IT A PITY...



Everything Isn't As Accommodating As

c-tree™ / **r-tree**™

FILE HANDLER — REPORT GENERATOR

Performance and Portability

For all the time you devote to developing your new programs, doesn't it make sense to insure they perform like lightning and can be ported with ease?

c-tree: Multi-Key ISAM Functions For Single User, Network, & Multi Tasking Systems

Based on the most advanced B+ Tree routines available today, **c-tree** gives you unmatched keyed file accessing performance and complete C Source Code. Thousands of professional C programmers are already enjoying **c-tree**'s royalty-free benefits, outstanding performance, and unparalleled portability.

Only FairCom provides single and multi-user capabilities in one source code package, including locking routines for Unix, Xenix, and DOS 3.1., for one low price! In addition, **c-tree** supports fixed and variable record length data files; fixed and variable length key values with key compression; multiple indices in a single index file; and automatic sharing of file descriptors.

r-tree: Multi-File Report Generator

r-tree builds on the power of **c-tree** to provide sophisticated, multi-line reports. Information spanning multiple files may be used for display purposes or to direct record selection. You can develop new reports or change existing reports without programming or recompiling and can use any text editor to create or modify **r-tree** report scripts including the complete report layout. At your option, end users may even modify the report scripts you provide.

Unlimited Virtual Fields; Automatic File Traversal

r-tree report scripts can define any number of virtual fields based on complex computational expressions involving application defined data objects and other virtual fields. In addition, **r-tree** automatically computes values based on the MAX, MIN, SUM, FRQ, or AVG of values spread over multiple records. **r-tree** even lets you nest these computational functions, causing files from different logical levels to be automatically traversed.

Unlike other report generators, **r-tree** allows you to distribute executable code capable of producing new reports or changing existing reports without royalty payments, provided the code is tied to an application. Your complete source code also includes the report script interpreter and compiler.

How To Order

Put FairCom leadership in programmers utilities to work for you. Order **c-tree** today for $395 or **r-tree** for $295. (When ordered together, **r-tree** is only $255). For VISA, MasterCard and C.O.D. orders, call 314/445-6833. For **c-tree** benchmark comparisons, write FairCom, 2606 Johnson Drive, Columbia, MO 65203.

Complete C Source Code & No Royalties!

Xenix is a registered trademark of Microsoft Corp. Unix is a registered trademark of AT&T.

CIRCLE NO. 119 ON READER SERVICE CARD

"How to protect your software by letting people copy it."

By Dick Erett, President of Software Security



Inventor and entrepreneur, Dick Erett, explains his company's view on the protection of intellectual property.

"A crucial point that even sophisticated software development companies and the trade press seem to be missing or ignoring is this:

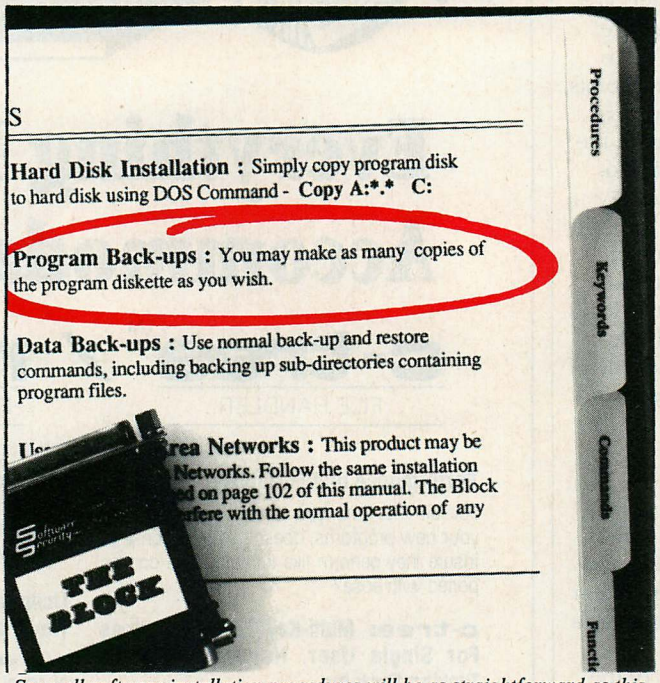
Software protection must be understood to be a distinctively different concept from that commonly referred to as copy protection.

Fundamentally, software protection involves devising a method that prevents unauthorized use of a program, without restricting a legitimate user from making any number of additional copies or preventing program operation via hard disk or LANs.

Logic dictates that magnetic media can no more protect itself from misuse than a padlock can lock itself.

Software protection must reside outside the actual storage media. The technique can then be made as tamper proof as deemed necessary. If one is clever enough, patent law can be brought to bear on the method.

Software protection is at a crossroads and the choices are clear. You can give product away to a segment of the market, or take a stand against the theft of your intellectual property.



Soon all software installation procedures will be as straightforward as this. The only difference will be whether you include the option to steal your product or not.

"...giving your software away is fine..."

We strongly believe that giving your software away is fine, if you make the decision to do so. However, if the public's sense of ethics is determining company policy, then you are no longer in control.

We have patented a device that protects your software while allowing unlimited archival copies and uninhibited use of hard disks and LANs. The name of this product is The BLOCK™.

The BLOCK is the only patented method we know of to protect your investment. It answers all the complaints of reasonable people concerning software protection.

In reality, the only people who could object are those who would like the option of stealing your company's product.

"...eliminating the rationale for copy-busting..."

Since The BLOCK allows a user to make unlimited archival copies the rationale for copy-busting programs is eliminated.

The BLOCK is fully protected by federal patent law rather than the less effective copyright statutes. The law clearly prohibits the production of work-alike devices to replace The BLOCK.

The BLOCK attaches to any communications port of virtually any microcomputer. It comes with a unique customer product number programmed into the circuit.

The BLOCK is transparent to any device attached to the port. Once it is in place users are essentially unaware of its presence. The BLOCK may be daisy-chained to provide security for more than one software package.

Each software developer devises their own procedure for accessing The BLOCK to confirm a legitimate user. If it is not present, then the program can take appropriate action.

"...possibilities... limited only by your imagination..."

The elegance of The BLOCK lies in its simplicity. Once you understand the principle of The BLOCK, hundreds of possibilities will manifest themselves, limited only by your imagination.

Your efforts, investments and intellectual property belong to you, and you have an obligation to protect them. Let us help you safeguard what's rightfully yours. Call today for our brochure, or a demo unit."

Software Security inc.

870 High Ridge Road Stamford, Connecticut 06905

203 329 8870

CIRCLE NO. 189 ON READER SERVICE CARD

a short section with examples. An excellent, separate index is provided for the CodeView section in the manual.

The demonstration disk, "Learning Microsoft CodeView," provides a very good overview to the product and shows some of the ways that it can be used. The demonstration disk is currently available separately from Microsoft and can be freely copied and distributed. It is recommended.

Microsoft provides a technical support hotline at no extra cost, and telephone support for CodeView is good. Microsoft technical support people are able to answer questions quickly.

As a full-featured visual debugger, CodeView is professional and solid. It can handle source code and assembly language in all memory models furnished by Microsoft C. In addition, the interface for third-party debugging hardware can be used to enhance the performance of CodeView's watchpoints and tracepoints.

As with any product, CodeView can be improved. Documentation for the display expression and dual-display features should be clarified. A method for presetting watch expressions before a procedure is reached would be a welcome addition. Compatibility with memory-resident programs should be added. CodeView is not friendly to pop-up programs such as Borland's SideKick. During testing, calling SideKick from within CodeView caused the AT to freeze up, requiring a reboot.

Features such as the execute mode and watch expressions provide quite a powerful method for monitoring a program's behavior in detail at the source code or assembly level. CodeView's features for source-level debugging make it much less intimidating to the programmer who is not familiar with assembly language or machine instructions. While providing debugging at source level, CodeView has not sacrificed the ability to perform the more traditional low-level debugging.

Microsoft Corporation
16011 Northeast 36th Way
P.O. Box 9098017
Redmond, WA 98073-9717
206/882-8080; 800/426-9400
C Compiler: $450
CIRCLE 343 ON READER SERVICE CARD

FORTRAN 4.0 Compiler: $450
CIRCLE 344 ON READER SERVICE CARD

Mark S. Ackerman is a software engineer currently working on Project Athena at the Massachusetts Institute of Technology.



The best new programs from one of the best-known publishers.

C++

ADVANTAGE C++

- New object-oriented language lets you develop large and complex programs more productively.
- Write reliable, reusable code that is easier to understand and maintain.
- Fully compatible with existing C programs and tools.
- All the benefits of C without its limitations.
- Available for Lattice and Microsoft C compilers.

Latest From LIFEBOAT

MULTITASKING

TimeSlicer

- Create multitasking and real-time applications in C now!
- No need to interface with the OS. Tasks can be created, suspended or terminated at run-time.
- Optimize processor usage and transparency.
- Compatible with Lattice C, Microsoft C, ADVANTAGE C++ and object-oriented programming.
- Header files for C and assembly language.

Latest From LIFEBOAT

LINKER

ADVANTAGE Link

- Fastest, most powerful PC-DOS overlay linker.
- First linker to take full advantage of EMS.
- Supports 53 commands.
- Compatible with Microsoft CodeView and Pfix Plus.

Brand New LIFEBOAT

GRAPHICS

ADVANTAGE Graphics

- Complete C programming graphics library.
- Create bar and pie charts with one command.
- No royalties or run-time fees.

Brand New LIFEBOAT

C COMPILER

ADVANTAGE C

- Newest technology based on AI and advanced design.
- Unmatched execution speed; full control of compilation environment.
- Advanced optimization routines.
- Over 250 library functions; full ANSI C library.
- Microsoft C compatibility.

Brand New LIFEBOAT

We make the best software even better.

After 10 years of publishing software, we know what's important to you. We're committed to full service that goes beyond just selling you the best software at competitive prices. Our expert staff can help you choose the right programs and provide full technical support. Count on Lifeboat for the complete solution to all your programming needs.

Call for the latest products from:

Blaise • Computer Innovations • CompuView • Essential • GSS • Gimpel • Greenleaf • Informix • Lattice • Media Cybernetics • Microsoft • Oregon Software • Periscope • Phar Lap • Phoenix • Rational Systems • Roundhill • SoftCraft • and more

Call **1-800-847-7078**
In NY **914-332-1875**
or your local Lifeboat Authorized Dealer.

The Full-Service Source for Programming Software.

LIFEBOAT

55 South Broadway
Tarrytown, NY 10591
Telex #510-601-7602

INTERNATIONAL SALES OFFICES

Australia/New Zealand: Charlton Distributors Phone: (64) (09) 766-361
Canada: Scantel Systems Phone: (416) 449-9252
England: Grey Matter, Ltd. Phone: (44) 364-53499
System Science, Ltd. Phone: (44) (01) 248-0962
France: Compusol Phone: (45) 30.07.37
Italy: Lifeboat Associates Italia Phone: (02) 464601
Japan: Lifeboat, Inc. Phone (03) 293-4711
Spain: Micronet, S.A. Phone: (34) 1-262-3304
The Netherlands: SCOS Automation BV Phone: (31) 20-10 69 22
West Germany: MEMA Computer GmbH Phone: (69) 34-7226
Omnitex Phone: (76) 23-61820

CIRCLE NO. 173 ON READER SERVICE CARD



9:02: *"A database manager? How many months will this take?"*



12:04: *"Wow, my first report. And there's still time for lunch."*



2:19: *"From now on I'm taking the easy way out."*

What it does is hard. How it does it is easy.

The reason most people use a database manager is to make work easier. But most database managers are so difficult to use they end up making work harder.

That's why many companies that had already standardized on complex systems like dBASE® are re-standardizing on DataEase.™

DataEase. No database manager that does as much is easier.

DataEase is easy because you don't have to program. Menus with easy to understand commands like "cut" and "paste" lead you through the process of setting up forms, files and reports.

And DataEase is just as powerful as much more complex systems. But the power of DataEase becomes available as you need it. So the transition from building simple applications to building complex ones is smooth. And easy.

DataEase also makes it easy to use information stored on other computers. Because the only thing that changes is where the data comes from. DataEase LAN™ and DataEase Connect™ automatically get the information you need. No matter where it is—on another PC, LAN server, minicomputer or mainframe.

Creating complex business graphics is easy with DataEase GrafTalk.™ And DataEase Developer™ has everything you need to document and encrypt your applications and build demonstration disks. And, of course, it's easy too.

The DataEase Family. They can do everything. And everything they do is easy.

Send us the coupon for more about the DataEase Family, including a free DataEase demonstration disk. Or call 1-800-334-**EASE**, or 203-374-8000.

PCTJ 3/87

DataEase International, Inc.
12 Cambridge Drive, Trumbull, CT 06611
1-800-334-**EASE**

☐ Please send me more information about
☐ DataEase ☐ LAN ☐ Connect
☐ GrafTalk ☐ Developer
☐ Please send a free DataEase demonstration disk.
☐ Please have a DataEase representative call.

Name_____ Title_____
Company_____ Phone_____
Address_____
City_____ State_____ Zip_____
No. of PCs on site ☐ 1-10 ☐ 11-50 ☐ 51+
☐ IBM PC/XT/AT ☐ IBM COMPATIBLES ☐ WANG ☐ DEC ☐ OTHER

Software Solutions, Inc. is now DataEase International, Inc.

DATAEASE™

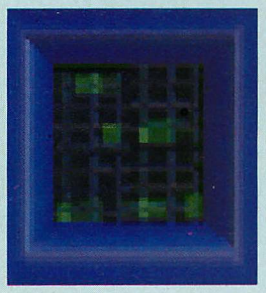
If it was hard to use we would have called it DataHard.

Canada (416) 231-1270 • United Kingdom 01-554-0582 • Italy 2-836-0097 • Benelux (31) 73-414855 (Holland) • Scandinavia (47) 71-46166 (Norway)
Switzerland 985152 • West Germany/Austria 89-461-3257 (Munich) • Australia (03) 699-7255 • Brazil 11-881-0600

dBASE is a trademark of Ashton-Tate.

CIRCLE NO. 144 ON READER SERVICE CARD

©1987 DataEase International, Inc.



Mapping PC Address Space

AUGIE HANSEN

Memory and I/O address maps can help developers identify the available memory and port locations for application programs.

The Intel 8086 family of microprocessors, including the 8088 and 80286 chips that are the backbones of the IBM PC product lines, cannot operate in a vacuum. They must communicate with entities outside themselves to perform useful work: memory, both RAM and ROM; and external devices such as display systems, disk drives, and keyboards.

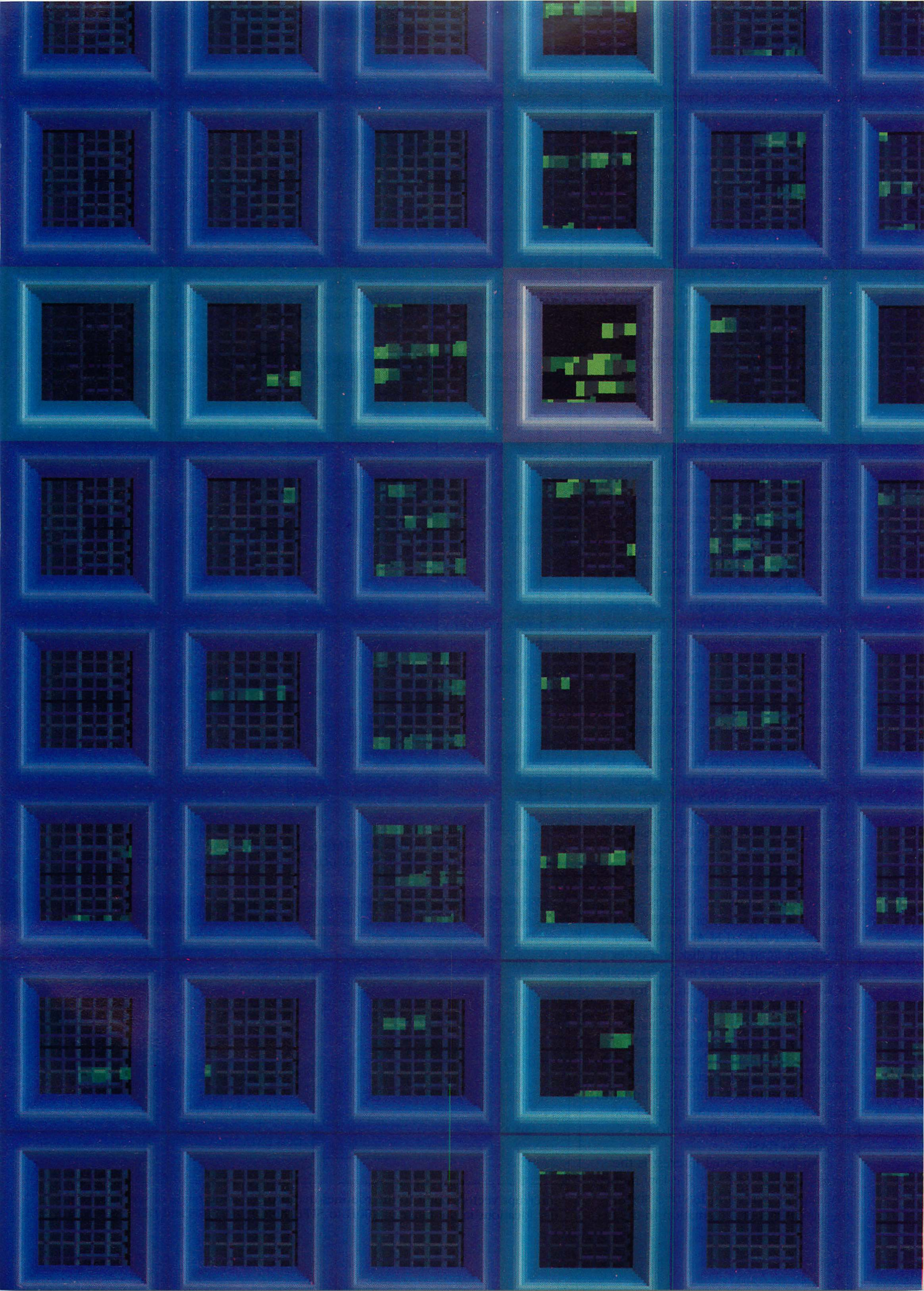
The microprocessor accesses memory and external devices in two primary ways. It uses a set of address leads to form either memory addresses or I/O addresses (called ports). The 8086/88-based machines have 20 address leads, permitting direct addressing of up to 1MB of memory. Figure 1 shows the basic 20-bit address bus used by the PC, PC/XT, and PC*jr*. Machines based on the 80286 use an additional 4 address leads for a total of 24, yielding a maximum directly addressable memory of 16MB.

All of the 8086 family microprocessors are also capable of addressing up to 64K ports. This separate address space is accessed via some of the same address leads used to access primary memory, but the processor operates in a different mode when reading and writing ports. The microprocessor can access either memory or a port, but not both at the same time. Instructions such as MOV use the address leads to access memory, but IN and OUT instructions cause the processor to access ports in the I/O address space. IBM chose to use only the first 10 address leads (A0-A9) when addressing ports, so PCs nominally have access to only 1,024 ports.

Certain areas of the memory and I/O spaces of a PC are reserved for system uses. The uses are, for the most part, common across the PC family, with some variations. One of the problems faced by developers is knowing which memory locations and ports are used by a given machine and which are available for use by application programs. The PC, PC/AT, XT, and PC*jr* (and most compatibles) can be treated identically for certain classes of system data and access points, but variations have been introduced that must be accounted for in program designs.

The memory and I/O address maps in this article are designed to aid programmers and hardware designers by making memory and port information readily available. The maps provide an overview of the allocation of memory and I/O address spaces in the PC machines. Areas of particular interest are expanded to show additional detail.

COMPUTER GRAPHIC • DOV JACOBSON



Only the first 640KB of the 1MB of address space can be populated with RAM; this is the maximum amount supported by current versions of DOS. The rest of the address space is dedicated to system-related items, such as ROM BIOS, BASIC, expansion ROM for add-in boards, and video display memory. In the user area of memory, system data and interrupt vectors, device drivers, and the operating system (which grow with each new release) occupy a significant amount of space. Add to that a generous amount of space for the omnipresent resident utilities and a user-friendly shell or task switcher, and it is a wonder that any room is left for application programs to load and run, even on machines that are equipped with the full complement of memory.

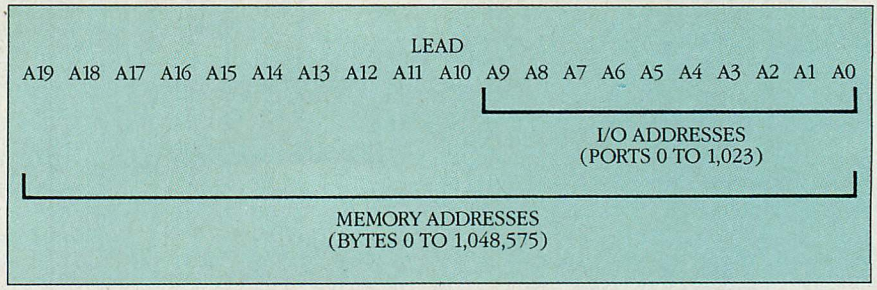
Figure 2 is a coarse memory address map showing the basic memory allocations envisioned by IBM. (All of the maps in this article use the convention of increasing addresses moving down the page and from left to right.) To run DOS, the primary RAM area must start at 0, be contiguous, and be large enough to run the operating system and its extensions (device drivers and other resident code) and to load and run the user's application program code and data. Isolated blocks of memory are not accessible to DOS except through special programs (RAM disk, print spooler, and so on), which are usually implemented as DOS device drivers or terminate-and-stay-resident (TSR) programs.

BASIC (except in the PC*jr*, which uses cartridge BASIC) and the ROM BIOS reside at the high end of the PC's memory space. The ROM BIOS is the firmware portion of the PC's basic input/output system, the low-level interface between the computer and devices that serve the user.

Versions of the ROM BIOS starting with the introduction of the XT use a feature called ROM SCAN to look for additional ROM (hard disk and other special BIOS modules) in the expansion ROM area. This enables an XT, for example, to boot from a hard disk instead of diskette drive A:. Any executable code found in the expansion ROM area is given control after the primary bootstrap sequence completes. Control eventually returns to the BIOS initialization routine, which either loads DOS (or another operating system) or defaults to BASIC.

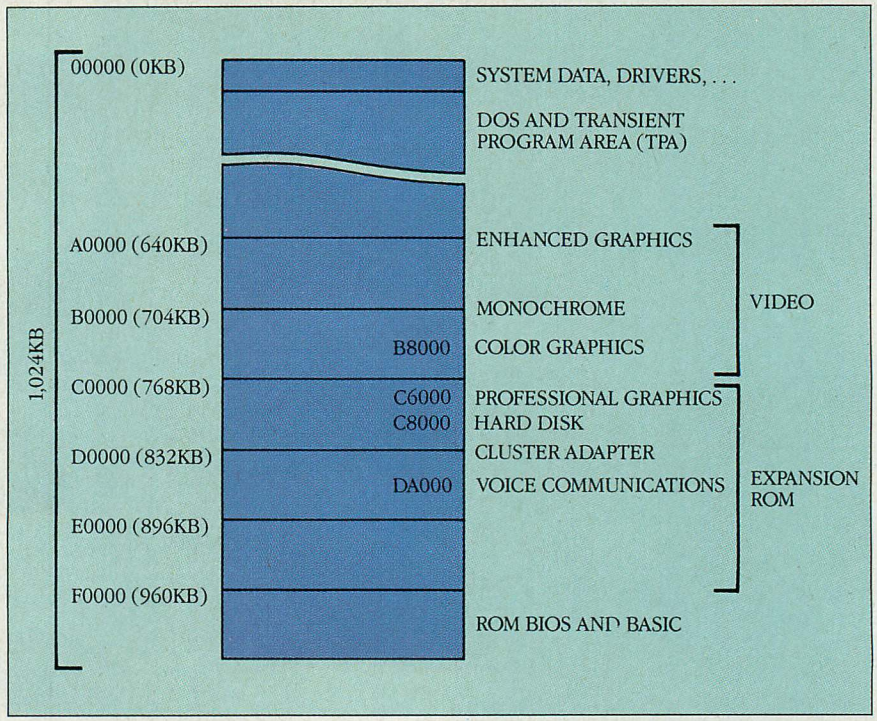
An area of special interest is the lowest portion of RAM. The low memory map presented in figure 3 shows the groupings of information in the re-

FIGURE 1: *PC Addressing*



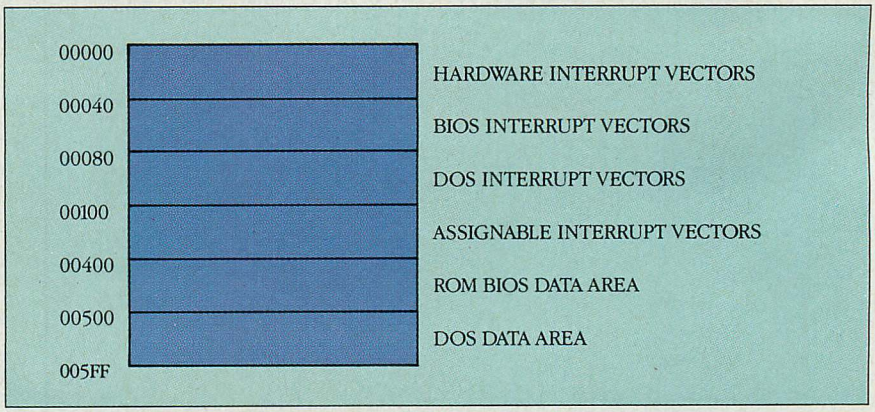
PCs based on the 8088 microprocessor use a 20-bit address bus. Machines based on the 80286 use a 24-bit address bus, yielding a 16MB address space.

FIGURE 2: *Coarse Memory Address Map*



Only the first 640KB of the PC's 1MB address space can be populated with RAM; the rest is dedicated to ROM, display buffers, and other system-related items.

FIGURE 3: *Low Memory Address Map*



Interrupt vectors point to the routines that provide wide range of services. Memory locations in the range of 400H to 5FFH hold ROM BIOS and DOS-related data.

gion from address 0H to 5FFH. The interrupt vectors point to the routines that provide a wide range of services on demand. Memory locations in the range of 400H to 5FFH hold ROM BIOS and DOS-related data. These locations are useful because they contain system configuration data, vital disk, keyboard, and video information, and miscellaneous status flags and values.

Some of the information, such as equipment flags, in the BIOS data area, shown in figure 4, is available with DOS and BIOS interrupts and should be obtained in that way to guarantee portability among PC models and versions of DOS. Other information, however, can be obtained only by reading locations directly. For example, two bytes provide keyboard status information. The first is at 417H. The value it holds is bit-significant to indicate the status of special keys, such as Caps Lock and Num Lock. The information can be obtained by calling BIOS interrupt 16H, function number 2 (Shift status). A second byte at 418H indicates whether one of the special keys is currently in a pressed or released state, plus some PC*jr*-specific key information. This information is not available through BIOS calls and must be read directly.

Reading information directly usually is not a recommended procedure. However, IBM says that it intends to maintain the structure and organization of the BIOS data area in its future PCs, so that programs can safely access the locations without becoming obsolete because of new equipment and operating system versions.

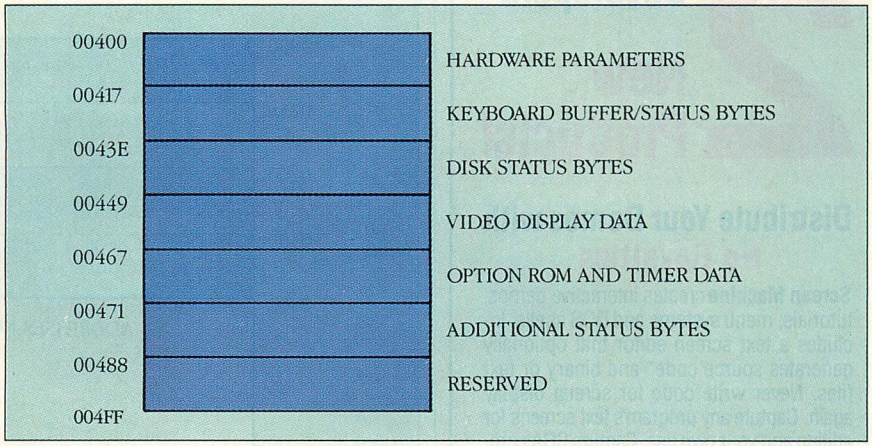
Above the transient program area (TPA) beginning at memory address A0000H is a reserved area for video adapters. Figure 5 shows the memory allocations for various video adapter configurations. Only a small sliver, 4KB in all, beginning at address B0000H is occupied by the standard IBM Monochrome Display and Graphics Adapter. The Color Graphics Adapter (CGA) starts at B8000H and extends upward for 16KB. Both a monochrome adapter and a CGA can be installed in a PC, but programs that depend on the standard video services provided by DOS and ROM BIOS can access only one display at a time. Programs that write to both display types simultaneously are using direct-screen access methods that are potentially troublesome to some PC operating environments (TopView, Windows, GEM). Quarterdeck's DesqView does not seem to be affected.

The region from A0000H to AFFFFH, although marked RESERVED by IBM, was used by some vendors as an extension of the TPA; DOS can be instructed to use RAM that is installed there provided it is contiguous with the lower regions of RAM. Several cards place an isolated block of memory in the region for use as a print spooler or RAM disk. These rogue memory allocations now conflict with applications that expect to find an Enhanced Graphics Adapter (EGA) starting at A0000H.

The EGA can contain up to 256KB of memory mapped into planes that can be allocated in several ways, (see "The EGA Standard", John T. Cockerham, October 1986, p. 48). When the EGA is masquerading as either a monochrome adapter or a CGA, its memory allocation starts at the expected monochrome adapter or CGA addresses and occupies 32KB. When configured to operate in pure EGA modes, the memory allocation starts at A0000H and occupies either 64KB or 128KB depending on the selected buffer address option.

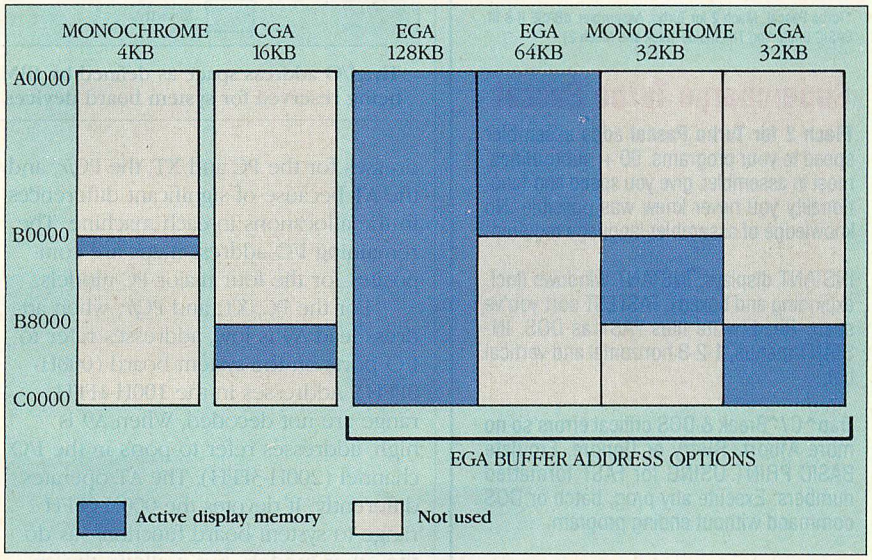
The maps specified by the overview in figure 6 show the I/O addresses that are currently allocated and reserved for future allocation by IBM. They are designed to help developers understand the mystery of the PC's I/O address space. Each row in a map (figures 7-12) represents 16 I/O addresses (ports), starting at the number shown at the left end of the row. Each figure maps 256 consecutive I/O addresses. The first 3 I/O address maps present information about the same block of ad-

FIGURE 4: *BIOS Data Area Memory Map*



Some of the information in the BIOS data area is available via DOS and BIOS interrupts and should be obtained in that way in order to guarantee portability.

FIGURE 5: *Display Memory Allocations*



The 128KB area beginning at memory address A0000H is reserved for video adapters. Only 4KB is occupied by the standard IBM monochrome display adapter.

2 Programmers & Developers New Products!

Distribute Your Demos with No Royalties

Screen Machine creates interactive demos, tutorials, menu systems and DOS shells. Includes a text screen editor that optionally generates source code* and binary or text files. Never write code for screen display again. Capture any program's text screens for editing and your own use. Capture CGA compatible graphics screens for BLOAD or direct display. SAVE hundreds of HOURS of work.

Now there's no need for separate screen and demo software packages and no need to pay outrageous royalties. Priced at only $79.00.

*Turbo Pascal, Mach 2 for Turbo, Assembler, dBASE II & III, BASIC (including The Inside Track and Mach 2).

Supercharge Turbo Pascal

Mach 2 for Turbo Pascal adds assembler speed to your programs. 90 + subroutines, most in assembler, give you speed and functionality you never knew was possible. No knowledge of assembler language required.

INSTANT displays. INSTANT windows (incl. exploding and boxed). FASTEST sort you've seen. Read/write files FAST as DOS. INSTANT menus, 1-2-3 horizontal and vertical bar.

Trap^C/^Break & DOS critical errors so no more A)bort, R)etry or I)gnore. Emulate BASIC PRINT USING for FAST formatted numbers. Execute any prog, batch or DOS command without ending program.

Read environment. Read file directory. Get/set file attributes. Plus too many string functions to describe here. No royalties when you distribute COM programs. All source code included. A true bargain at $69.00.

NOT COPY PROTECTED. 30 Day Money-Back Performance Guarantee. Requires IBM/compatible & DOS 2 +.

Order Now 800-922-3383

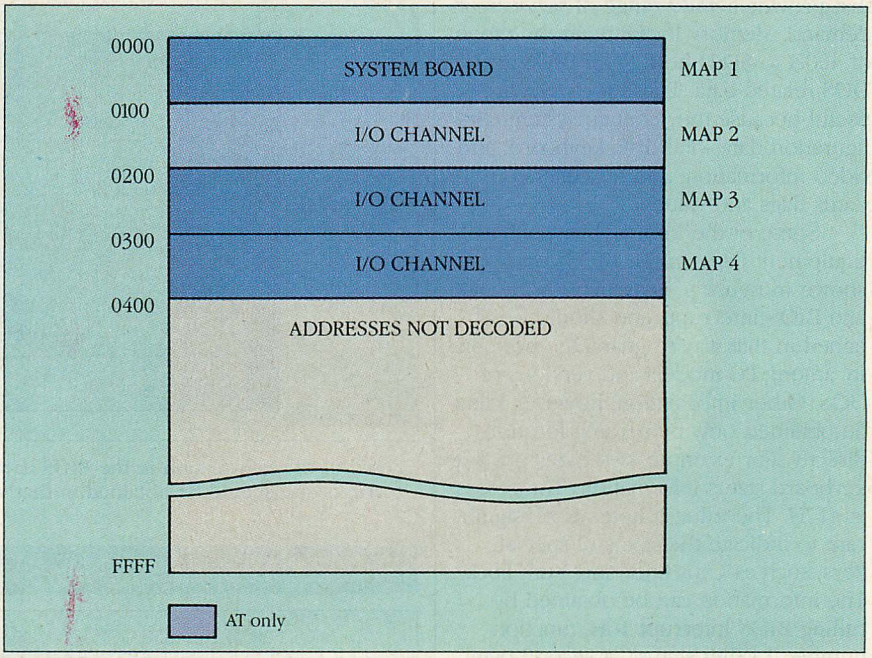
We welcome VISA/MC. COD US only $3. S/H US $3, Canada $5, Elsewhere $18. GA res. add tax and call 404-973-9272. Demo available. Send $5 check. Refunded on direct purchase.

We also publish Stay-Res, Mach 2 for BASIC, The Inside Track and Peeks 'n Pokes.

MicroHelp, Inc.
2220 Carlyle Drive
Marietta GA 30062

CIRCLE NO. 257 ON READER SERVICE CARD

FIGURE 6: *PC I/O Address Space*



The I/O address space as defined by IBM is divided into several areas, with the first being reserved for system board devices and the remainder for I/O channel devices.

dresses for the PC and XT, the PC*jr*, and the AT because of significant differences in the allocations in each machine. The remaining I/O address maps are composites for the four major PC models.

For the PC, XT, and PC*jr*, when address lead A9 is low, addresses refer to I/O ports on the system board (000H-0FFH). Addresses in the 100H-1FFH range are not decoded. When A9 is high, addresses refer to ports in the I/O channel (200H-3FFH). The AT operates differently. It devotes the 000H-0FFH range to system board functions, as do the other models, but it allows the entire 100H-3FFH range to be used by adapters in the I/O channel. To date, the only allocation by IBM in the 100H-1FFH block of addresses is for the AT's hard-disk controller at 1F0H-1F8H.

Three separate maps are presented for the 000H-0FFH range. Figure 7 shows PC and XT allocations; figure 8 shows only PC*jr* allocations; and figure 9 shows only AT allocations. The AT claims nearly all of the I/O addresses, having both enlarged allocations for various support chips (timer, DMA, PIC) and added allocations for a second interrupt controller, a second DMA controller, the realtime clock, and the numeric coprocessor. CMCB at I/O address 0F0H is *Clear Math Coprocessor Busy* and RMC at 0F1H is *Reset Math Coprocessor*. A composite of the three maps reveals a potential problem for IBM; no free I/O ports are available in the block assigned to system board use. Future PC designs may need to expand the I/O address allocations and assign system uses to areas previously reserved for external devices in the I/O channel.

The 100H-1FFH range (figure 10) is used only in the AT and only for the hard disk. The map, therefore, shows many open I/O addresses in this range. Hardware that depends on the use of ports in this address range, however, functions properly only in an AT or a compatible machine. Consequently, for compatibility across the PC family, hardware designers should try to avoid this range of I/O addresses.

The 200H-2FFH address range (figure 11) appears to have a lot of open space, but it is actually a crowded range. Communications ports, special graphics interfaces, and hundreds of other optional hardware devices are all vying for very limited real estate in this area. A full 16 ports have been reserved for game I/O, although to date, systems use only port 201H. When an expansion unit is added to a PC, the 8 ports starting at 210H are used for communications between the units.

IBM is beginning to use the block of 48 RESERVED ports starting at port 220H. The voice communications adapter sits at port 21FH. (Three alternative assignments are above 3FFH.)

The block of I/O addresses starting at 278H is reserved for use by a parallel printer. If the system is equipped with a

GOODBYE, IBM...



THANKS FOR THE MEMORIES

Have you ever noticed how time seems to change everything? Fashions change. Technology changes. Even the microcomputers we use change. Maybe that's why so many of our customers are saying goodbye to IBM in favor of a "good buy" from Wells American.

As time goes by, more and more computer users are realizing what an extraordinary value our A★Star II® truly is. It's the *only* AT class microcomputer that can run at 6, 8, 10 *and* 12 MHz! It's also the *only* PC/AT compatible that's "network ready." Better yet, each A★Star II now comes with your choice of keyboards - the original AT version or the enhanced "RT" style. Best of all, A★Star II prices start at only $995!

But what about quality and support? Don't worry! We've been making microcomputers *longer* than IBM! And it shows. Our A★Star® computers have been top rated by leading industry trade journals. Even so, we've heard that some "big blue" old-timers still worry about trading their IBM "security" blanket for better priced, higher performance equipment. (Remember how difficult it was to give up *your* baby blanket?) That's why we've arranged for RCA, one of the world's largest technical service organizations with 18,000 employees, to provide low cost, nationwide maintenance for our A★Star II. And if that's not enough, every unit includes free schematics and a no risk money-back guarantee.

THE A★STAR II IS MADE IN THE USA.



Still need a blanket? We've got you covered! Try out one of our A★Star II's and we'll send you one. It's baby blue, monogrammed and...it's absolutely free! We'll also include a $395 option, at no charge, with your first A★Star II purchase. This offer *is* limited, so call or write us today. Just because you'll be saying goodbye to IBM doesn't mean you'll have to say goodbye to quality, support *or* security. Besides, at least you'll still have your memories...and your money!

Wells American®

Corporate Headquarters: 3243 Sunset Boulevard • West Columbia, South Carolina 29169 • 803/796-7800 • TWX 510-601-2645

IBM, Personal Computer AT and AT are trademarks of International Business Machines.

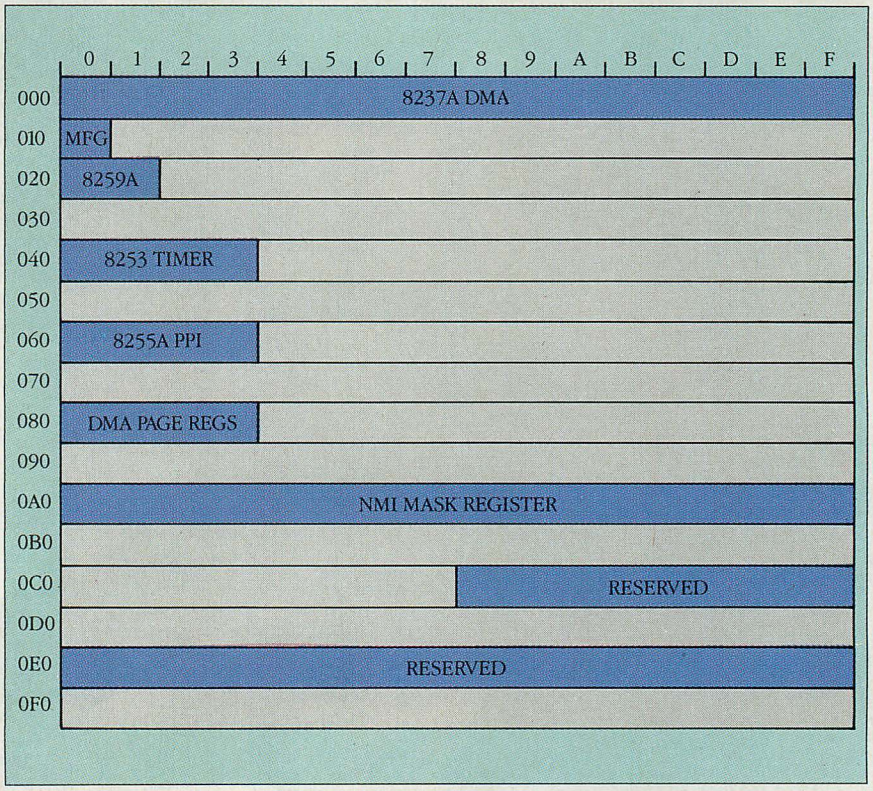
CIRCLE NO. 124 ON READER SERVICE CARD

monochrome adapter, this becomes LPT3; otherwise, it is treated as LPT2. The initialization phase of the DOS loading procedure accommodates the hardware configuration automatically. The secondary asynchronous communications adapter (COM2) is allocated eight ports beginning at 2F8H. Additional allocations may be made for serial adapters at other I/O addresses; however, a specially written device driver or application code is required to access the new, nonstandard allocations.

The range of I/O addresses from 300H-3FFH (figure 12) also is mostly accounted for with the various allocations for network and communications adapters and the EGA that have been added in the last year or two. This range was already heavily laden with assignments for video display hardware (monochrome adapter and CGA), communications devices of both the synchronous and asynchronous persuasions, and the hard-disk controller (except for the AT hard-disk controller that was placed in ports 1F0H-1F8H). The parallel printer assignment starting at 378H is LPT2 if an IBM monochrome display/printer adapter is installed or LPT1 if not.

Neither IBM nor any other organization is making an effort to parcel out the available I/O addresses in a coordi-

FIGURE 7: *System Board Ports, PC and PC/XT*



PCs require I/O addresses only for the interrupt controller, timer, and speaker driver system devices located on the system board; quite a bit of open space exists in this map compared to that of the AT.

Soft*Rite ⟩⟩ LANbasic!

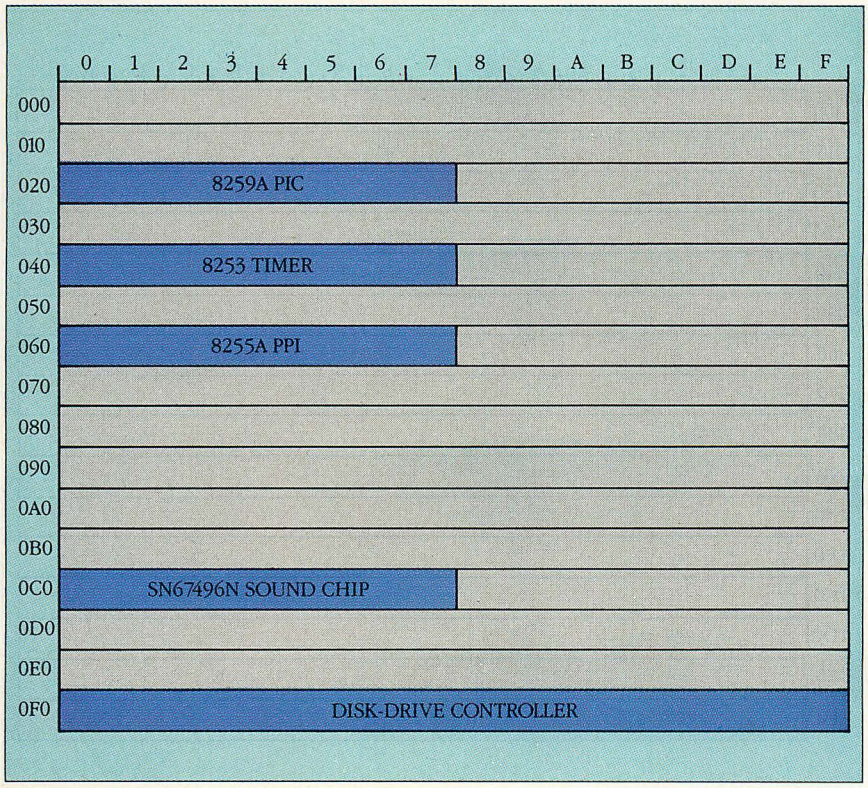
Soft*Rite announces a *Superior* three-part programmer's tool. MicroSoft BASIC™ compatible in every place that counts. *

LANscreen	LANbasic	LANdbase
LANscreen makes the burden of defining your database record structures one that you will look forward to instead of dread. Standard fields and types such as "Money", "Telephone" and "Date" are one keystroke to generate. As many as 255 database structures can be related to a single screen! Segmented data input can be done automatically by editing the field display. Part numbers with spaces or dashes and slashes are automatically parsed down to the essential raw data. Definable Upper and Lower limits for numeric inputs along with ACCEPT/EXCEPT input filters are standard features. Screen Mask generation is done with a "freestroke" approach. You may put mask elements and data input/output cells wherever you wish. COLOR them too! EGA support too! *Drivers Installed for IBM PC-NET/MS-NET	LANbasic is your own personal solution to powerfull data manipulation. How many times have you got excited over some new "total" database package only to find out (after spending a fair amount of time and money) that you were stuck in some corner, unable to do some function that has become standard in "**I**n **B**usiness for **M**oney's" Basic? The manual is 400 pages long, so we cannot fully describe all the features, but here are a few in ADDITION to the ones you are now used to: ★ COMDATA$ 14 common areas ALWAYS available to inside or outside, chained or linked programs ★ Re-assignable printer ports LPT1 -LPT4 ★ Generic filename use that allows file and database locations to be redefined outside of basic in a user-created REDIRECTOR file, to ease multi-user system configuation ★ USESCREEN, <1-16> ★ SCREENINPUT, <anyfield> ★ SCREENOUTPUT, <anyfield> ★ OPENDB, <remote or local database manager> ★ DBGET, <variable from DBM, automatically defines and dimensions in LAN basic> ★ DBPUT, <same>	LANdbase is the home for your data. LANbasic calls are coupled to LANdbase via network comunications (PCnet or ?). A single keyvalue and function number will return a record. Multiuser record locking is handled by simply putting an 'X' after the read call. (i.e. RDDBEQUX, <argument>. Automatic "health checking" to warn you of poor hardware performance and lost or fragmented data. "Paranoid" mode of operation where files not accessed for some time will be closed and reopened to flush buffers and insure integrity. Password, Userlevel and Data encryption functions. Several DBM's can be installed in the network system to improve performance and reliability. Toggle mode screen (printer) reporting to record log-on or other access activites. Bill Fairman's tried and proven true C-Tree(c) data management product.

Soft*Rite Multi-User Programming Tools
15381 Chemical Lane, Huntington Beach, CA. 92649
(714) 898-0525

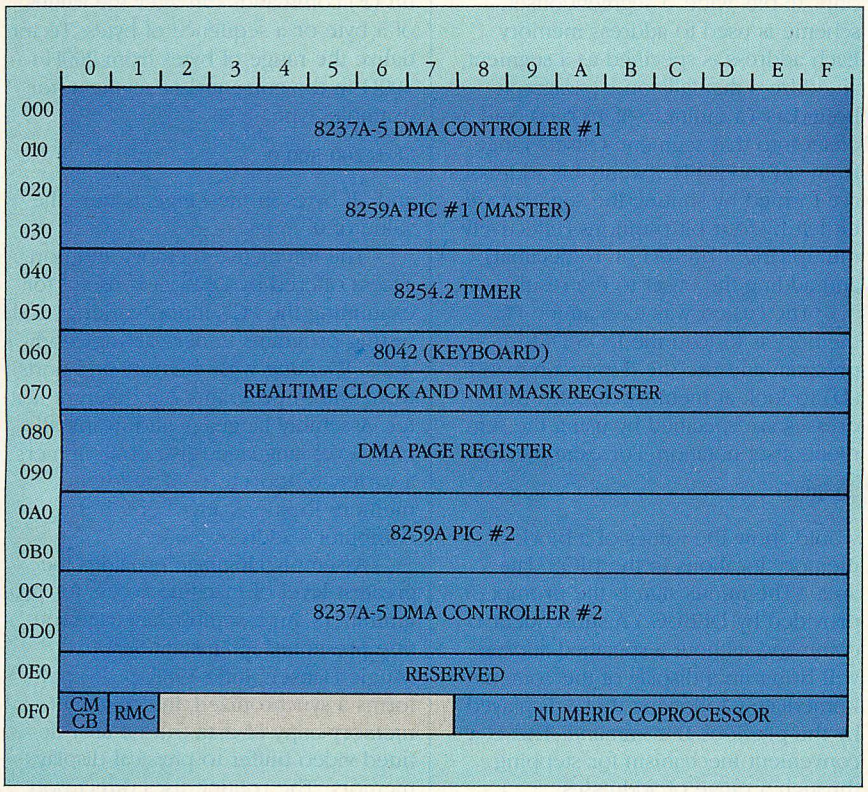
CIRCLE NO. 186 ON READER SERVICE CARD

FIGURE 8: *System Board Ports, PCjr*



In addition to addresses for system board devices common to the PC, the PC*jr* requires addresses for both its diskette drive and its sound generator.

FIGURE 9: *System Board Ports, PC/AT*



The AT claims nearly all of the system board I/O addresses, having both enlarged allocations for existing support devices and designated locations for new ones.

New Version 2.0

Complete C Programs in Half the Time, with *Instant-C*™

You can create programs much faster with *Instant-C* than with conventional programming tools. How? Because *Instant-C* is a high-performance interpreter, there are **no compile or link delays.** Change your program, then test it immediately. No matter how large your program, the turnaround time is just seconds.

"Instant-C means instant gratification."—*PC Magazine,* **Editor's Choice** for best C interpreter. *10/29/85*

Powerful **source-level debugging** saves your time. Conditional breakpoints, single-stepping by statement, source code backtraces, data monitoring, and many other debugging features make it easy to wipe out bugs quickly. Direct execution of any statement or function makes testing a breeze.

"The resulting debugging and testing capabilities are fantastic and the detailed trace/debug/display commands make it easy."—*The C Journal, Summer/85*

Instant-C checks pointer references for reasonableness, and checks that array indexes are within declared bounds. This **run-time checking** stops your program as soon as errors occur, for easiest debugging.

Not only does *Instant-C* help you quickly change, test, check and debug your code, but it runs your program **fast enough for real-time** applications.

"It is much faster than any of the other products mentioned and was the only one able to complete the standard SIEVE in a reasonable time. Clearly, this high speed allows much more complex problems to be attacked with Instant-C than with any of the other products discussed."—*Computer Language, 2/86*

Immediate feedback and precise diagnostics make *Instant-C* great for learning C. Full K&R and the ability to **link compiled object code and libraries** (Lattice and Microsoft) makes *Instant-C* compatible with your existing programs.

Instant-C makes all parts of the programming task as fast as possible.

"Clearly, Instant-C is the performance champion."—*PC Tech Journal, 5/86*

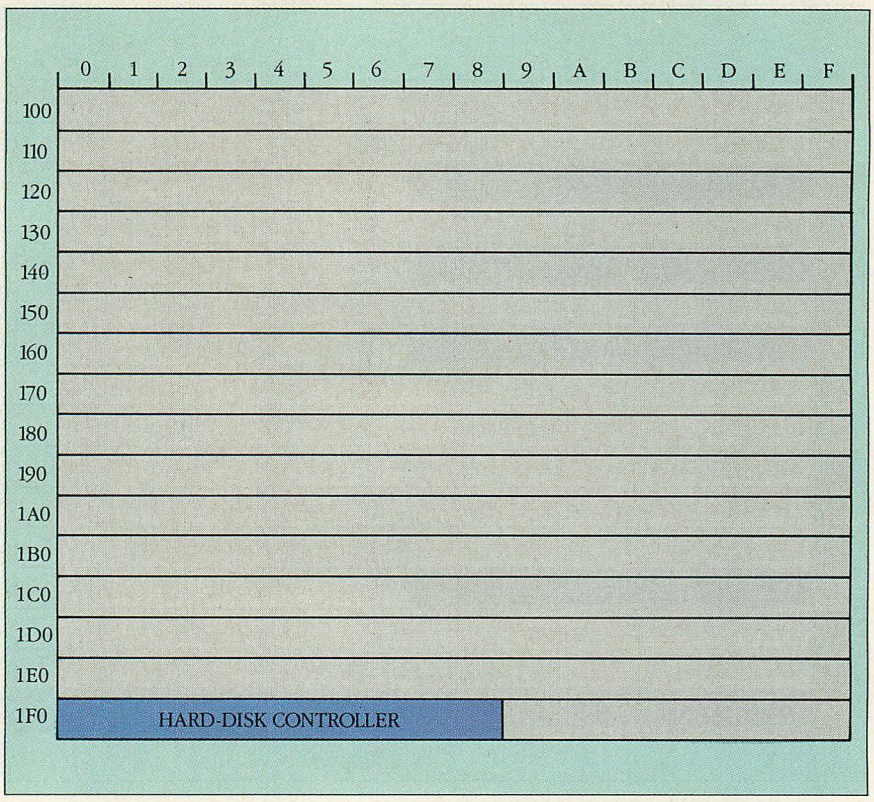
Version 2 works with MS-DOS and PC-DOS, and has a full 31 day **money back guarantee.** *Instant-C* is only $495. Order today! Call or write for full information.

Rational Systems, Inc.
P.O. Box 480
Natick, MA 01760
(617) 653-6194

CIRCLE NO. 181 ON READER SERVICE CARD

nated way. With the introduction of multiport serial boards (see "Beyond COM2," Augie Hansen, September 1986, p. 68), the crowding has worsened. Some of the multiport boards require 32 or even 64 contiguous I/O addresses, which makes it very difficult to find a home for them. This is why add-in boards that allow users a wide latitude in the selection of I/O addresses through switches or jumpers have definite advantages over comparable products that use fixed port assignments.

FIGURE 10: *I/O Channel Ports, PC/AT Only*



This range of I/O addresses is used only in the AT and only for the hard disk. Devices that depend on the use of these ports will not function properly on PCs.

EXPANDING THE SPACE

Adapter cards are free to do I/O address decoding using more than the ten lines decoded by the system board and by most of the earlier IBM adapters, thus permitting port numbers up to FFFFH if all 16 address lines are used. Recently introduced adapter cards are beginning to set the trend toward expanded I/O address space.

Expanded memory boards use two additional address lines to obtain larger allocations of I/O addresses (see "Expandable Memory," Ted Mirecki, February 1986, p. 66). IBM's Cluster Adapter can be set to decode addresses in the ranges of 790H-793H, 890H-893H, 1390H-1393H, or 2390H-2393H because it decodes two additional address leads. Some recent products, such as the serial boards made by ESE Technologies, use all 16 address lines to decode any of the 64K I/O addresses that can be presented on the address bus.

There is a catch, however. Any board in the I/O address space that decodes only the first 10 address lines will respond to any higher address, formed with a greater number of address lines, that happens to present the expected bit pattern on the lower 10 address lines. For this reason, developers still must be careful to avoid using previously claimed port numbers.

The advantage to using more address lines is that a given available port number, say 250H, can be multiplexed into as many as 64 separately addressable ports (one port specified by lines A0-A9 multiplied by the 64 patterns that can be specified by the upper 6 address lines). As more and more board manufacturers apply this technique, the port congestion problem will diminish a bit, at least in the near term.

ACCESSING MEMORY AND PORTS

The following examples show how to read and write memory and port values. Reading and writing PC memory would be easier if the Intel microprocessors had a linear address space. But because the registers that do the addressing are only 16-bits wide, a segment/offset scheme is used to address memory. Each address is specified as a segment, an address that falls on a paragraph boundary (a multiple of 16 bits), and an offset into that segment. Offsets range from 0H to FFFFH. Twenty-bit addresses are formed by shifting the segment value left by four bit positions (effectively multiplying it by 10H or 16 decimal) and adding the offset to the result.

The easiest way to examine PC memory is to load the DOS DEBUG program and use the dump command (**D**) to look at memory directly. Addresses are specified by using the segment:offset notation. For example,

```
-d 400:0
```

would show the values of a block of memory locations in the BIOS data area. (The minus sign is the prompt provided by DEBUG.) A subsequent **D** command without a specified address will bring up a display of the contents immediately following those displayed by the previous **D** command. This is a convenient mechanism for stepping through a range of addresses.

The DEBUG program also can be used to alter and write byte values anywhere in memory. The enter (**E**) and fill (**F**) commands can alter the values of a byte or a sequence of bytes. To initialize the range of bytes from 200H to 300H in the current data segment, for example, type

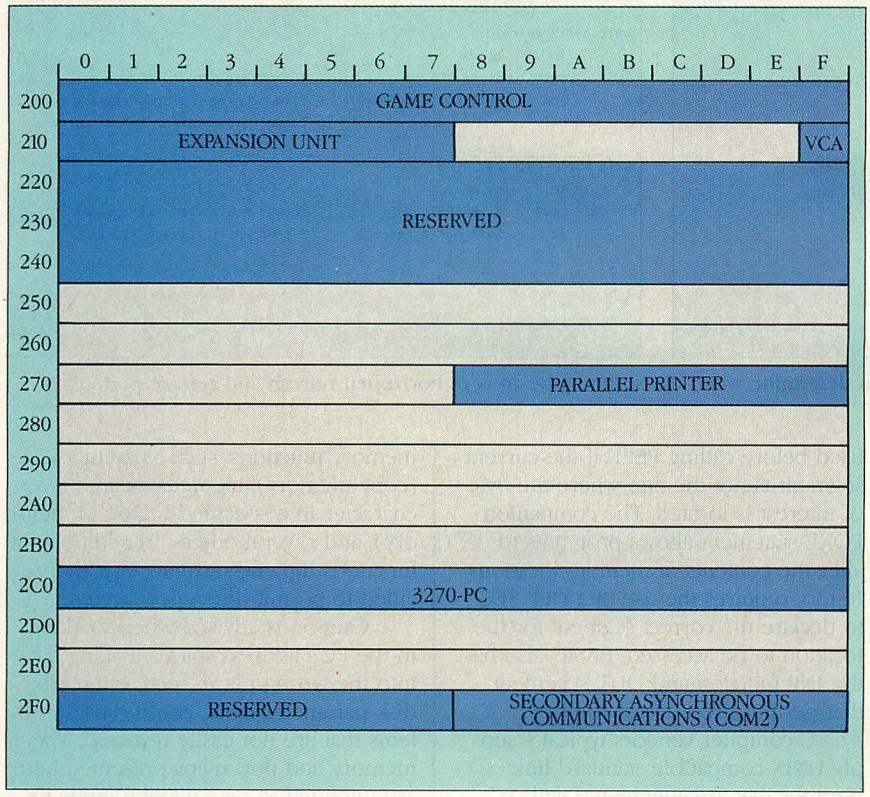
```
-f ds:200 300 0
```

and all bytes in the range assume the value of 0.

The interactive memory and port access offered by DEBUG is useful for examining the PC's inner workings and testing programs under varying conditions. Working with memory and ports from within programs is a different matter. Assembly language and nearly all high-level languages give programmers a variety of tools to read and write memory locations anywhere in the computer's address space.

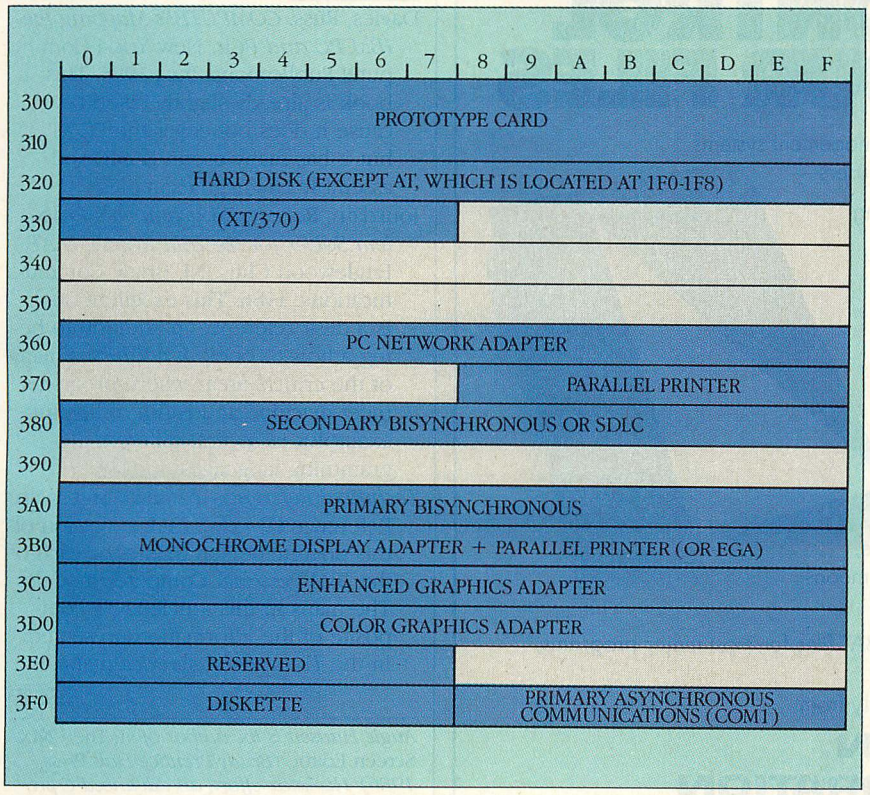
Assembly language provides the greatest level of machine access for programmers. An assembly language routine presented in "Instant Screens" (Augie Hansen, June 1986, p. 96) performs a synchronized, intersegment memory copy of data from a user-defined video buffer to physical display memory. The routine uses direct port access to handle critical timing tasks and a string move instruction to copy

FIGURE 11: *I/O Channel Ports, All PCs*



This range of I/O addresses is actually quite crowded. Hundreds of optional hardware devices, such as communications adapters, are all vying for ports here.

FIGURE 12: *I/O Channel Ports, All PCs*



This range of I/O addresses is also mostly accounted for by network and communications adapters and video adapters that have been added recently.

microDCF™
Text Workstation

IBM DCF3 Compatible Text Processing System For Your PC

- IBM Script language
- IBM GML language
- macro libraries
- symbol support
- multi-pass processing
- high performance
- laser printer support
- table of contents
- back-of-book index
- user-definable languages
- modular print manager
- and much more...

$795.-
Quantity Discounts
Site Licenses
Full Maintenance

APS/SPF™
Programmer Workstation

IBM ISPF Compatible Program Development Tools For Your PC

- ISPF PDF editor
- dialog manager
- dialog panel editor
- utilities
- multitasking while editing
- directory file selection
- customizable master menu

from $145.-

ALSI

Arrix Logic Systems Inc.
PO Box 142, Don Mills Stn.
Ontario M3C 2R6, Canada
(416) 292-6425
TELEX 06-986766 TOR
IBM is a registered trademark of International Business Machines Corp.

CIRCLE NO. 122 ON READER SERVICE CARD

FIGURE 13: *Assembly Language Port Access*

```
Listing 1.  SYNC.ASM
        .
        .
        .
;------ video status information ------
VSTAT   equ     3dah            ; video (CRT) status register
HRTRCE  equ     1               ; horizontal retrace bit mask
        .
        .
        .
        mov     dx,VSTAT                ; read c/g adapter status register
```

```
wait_horiz_refresh:
        in      al,dx
        test    al,HRTRCE               ; test horizontal retrace bit
        jnz     wait_horiz_refresh      ; loop until not in a retrace period
        cli                             ; can't tolerate an interrupt here
wait_horiz_retrace:
        in      al,dx
        test    al,HRTRCE               ; test horizontal retrace bit
        jz      wait_horiz_retrace      ; loop until retrace starts
        .
        .
        .
```

The CGA's control port can be examined to determine when the scanning beam is in horizontal refresh and retrace periods.

data quickly from the program data segment to the display data segment.

The essence of the routine is presented in the code fragment in figure 13, which shows how to examine the CGA's control port at 3DAH to determine when the scanning beam is in horizontal refresh and retrace periods. To avoid visual interference, CGA memory should be accessed only during horizontal or vertical retrace periods.

BASIC provides the PEEK function, which yields the byte value at the specified address (offset) in the current segment. A DEF SEG statement must be used before calling PEEK if the current segment is not the one where the byte of interest is located. The companion POKE statement allows programs to alter the contents of memory locations. It, too, requires the use of a DEF SEG to declare the correct segment for the location to be accessed. BASIC also has the INP function and OUT statement that permit direct access to ports.

C compiler vendors typically supply UNIX-compatible standard link libraries that are augmented with PC memory and port-access functions. Most C libraries provide a set of standardized memory functions, such as **memcpy** (copy memory), **memchr** (search for a character in a specified region of memory), and several others. In addition, the functions **inp** and **outp** are usually provided to permit direct port access.

Caution is advised when working in the PC's address spaces. Poking data into the wrong place, such as a hard-disk parameter table, could cause problems that are not easily reversed. The memory and port maps presented here are intended to be a general guide to where the important hardware elements of PCs are located and where new ones might be safely placed.

TOTAL CONTROL OF ALL YOUR FILES

EasyPath. The Cadillac of hard disk management systems
—Especially for programmers and power users—

EasyPath fills the void in DOS by allowing complete access to all your hard disk files.

EasyPath lets you

- Access any file from any subdirectory
- Redirect file retrieval or file storage
- Execute any program from any subdirectory
- Specify pathing for COM, EXE and all other files
- Locate any file anywhere on your hard disk
- Find files by name or attribute
- Change attributes of files
- Pipe files to RAM disks
- Search for or redirect files by name, extension, attribute, subdirectory or other means
- Use wildcards and global file names



EasyPath comes with pre-configurated BAT files for most major programs, a complete user's guide, and online help.
Now at a new low price of **$59.95** directly from:

ISOGON CORPORATION
330 Seventh Avenue, New York, New York 10001 212 967-2424

CIRCLE NO. 139 ON READER SERVICE CARD

REFERENCES

Davies, Russ. *COMPUTE!'s Mapping the IBM PC and PC*jr. New York: Compute! Publications, Inc., 1985. This book is already slightly dated because it does not cover the PC/AT, but it has many useful BASIC and assembly language examples.

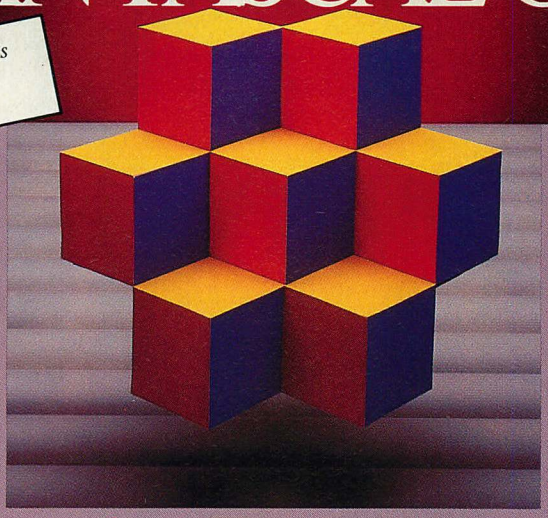
Jourdain, Robert. *Programmer's Problem Solver for the IBM PC, XT & AT.* Englewood Cliffs, NJ: Brady Communications, 1986. This excellent book is a desk reference in addition to being a how-to book. It addresses many of the critical areas where programs meet machine and is full of helpful examples in several different programming languages.

Technical Reference manuals (Hardware Reference Library), IBM. A version of this document is available for each of the IBM Personal Computer models. The maps in this article are a composite of the information presented in the *Technical Reference* manuals.

Augie Hansen is the author of vi: The UNIX Screen Editor, *(Brady/Prentice Hall Press, 1986). His latest effort, an advanced C programming book called* Proficient C, *is due out in March from Microsoft Press.*

WHY LOGITECH MODULA-2 IS MORE POWERFUL THAN PASCAL OR C.

"A clear winner... The integrated editor is a joy to use." BYTE Magazine, Jan. '87



APPRENTICE PACKAGE $99

WIZARDS' PACKAGE $199

- **Separate Compilation w/inter-module typechecking**
- **Native Code Generation**
- **Large Memory Model Support**
- **Most Powerful Runtime Debugger**
- **Comprehensive Module Library**
- **Maintainability**
- **Translator from Turbo and ANSI Pascal**

NEW!

APPRENTICE PACKAGE $99

Everything you need to begin producing reliable maintainable Modula-2 code. Includes the Compiler with 8087 support, integrated Editor, Linker, and BCD Module. We're also including FREE our Turbo Pascal to Modula-2 Translator!

NEW!

WIZARDS' PACKAGE $199

This package contains our Plus Compiler—for professional programmers or for those who just want the best. The Plus Compiler with Integrated Editor requires 512K and takes advantage of the larger memory to increase compilation speed by 50%. Our Turbo Pascal to Modula-2 Translator is also included at no extra charge.

NEW!

MAGIC TOOLKIT $99

We've put our most powerful development tools into one amazing Toolkit for use with either the Apprentice or Wizards' packages. Highlighted by our Runtime Debugger, the finest debugging tool available anywhere, the Toolkit also includes our Post Mortem Debugger, Disassembler, Cross Reference utility and Version which keeps track of different versions of one program. Our MAKE Utility figures out module dependencies and automatically selects those affected by code changes to minimize recompilation and relinking. We also provide source code of our major library modules for you to customize—or just play with.

WINDOW PACKAGE $49

Now you can build true windowing into your Modula-2 code. Features virtual screens, color support, overlapping windows and a variety of borders.

ROM PACKAGE AND CROSS RUN TIME DEBUGGER $299

For those who want to produce rommable code. You can even debug code running in ROM from your PC.

Turbo Pascal is a registered trademark of Borland International.

Call for information about our VAX/VMS version, Site License, University Discounts, Dealer & Distributor pricing.

To place an order call toll-free:

800-231-7717

In California:

800-552-8885

WIN A FREE TRIP TO Switzerland



HOMELAND OF MODULA-2

Return your Modula-2 Registration Card or a reasonable facsimile* postmarked between March 1, 1987 and May 31, 1987 to be included in a once-only drawing!

Grand Prize: One week excursion for 2 in Zurich, Switzerland including a guided tour of ETH, the University where Modula-2 was created by Niklaus Wirth. European customers may substitute a trip to Silicon Valley, California.

Second and Third Prizes: LOGITECH C7 Mouse or LOGITECH Bus Mouse with Paint & Draw software—a $219 value, absolutely free!

*Write to Logitech, Inc. for a registration card facsimile.

YES! I want the spellbinding power of LOGITECH Modula-2!

☐ Apprentice Package	$99
☐ Wizards' Package	$199
☐ Magic Toolkit	$99
☐ Window Package	$49
☐ ROM Pkg/Cross RTD	$299

Add $6.50 for shipping and handling. Calif. residents add applicable sales tax. Prices valid in U.S. only.

Total Enclosed $_____

☐ VISA ☐ MasterCard ☐ Check Enclosed

Card Number _____ Expiration Date

Signature

Name

Address

City _____ State

Zip _____ Phone

LOGITECH

LOGITECH, Inc.
805 Veterans Blvd. Redwood City, CA 94063
Tel: 415-365-9852

In Europe:
LOGITECH SA, Switzerland
Tel: 41-21-879656 • Telex 458 217 Tech Ch

In Italy:
Tel: 39-2-215-5622

CIRCLE NO. 229 ON READER SERVICE CARD

Modular Developments

Boasting high-level and low-level programming, separately compiled modules, and multitasking, Modula-2 is a language of the future. These six compilers bear out its potential for development.

JOHN T. COCKERHAM



To the field of computer languages, Niklaus Wirth has contributed three. His first, PL360, never really took hold as intended in the world of IBM 360 machines, yet within it Wirth had laid the groundwork for his next labor, the decidedly more successful Pascal. This second work continued a progression in design and technique that have culminated in a language that boasts strong typing and highly structured syntax, yet offers flexibility and low-level accessibility: Modula-2.

The language itself is reconsidered here (pursuant to its introduction in *PC Tech Journal* two years ago), and the current crop of Modula-2 compilers is appraised. (See "Modular Construction" and "Modular Implementation," Tom Woteki, Alan Frieden, Dov Levy, Thor Bestul, and Robert Stine, November 1984, p. 72 and December 1984, p. 154, respectively). The current list of language implementations includes Interface Technologies Corporation (ITC) M2SDS (and SDS-XP), Logitech, Inc. Modula-2, Modula Corporation PC Modula-2, PCollier Systems Modula-2PC, Pecan Software Systems Modula-2, and Workman & Associates FTL.

Wirth had introduced PL360 in 1968 as a single-pass, structured, high-level assembly language for use on IBM 360/370 mainframe machines. Some of the features of Pascal were evident in its forebear. An assignment statement had an ALGOL-like appearance—`r1 := r1 + 1;` meant add 1 to register 1, and FOR loops used the 360 registers as operands and compiled directly to hardware looping instructions.

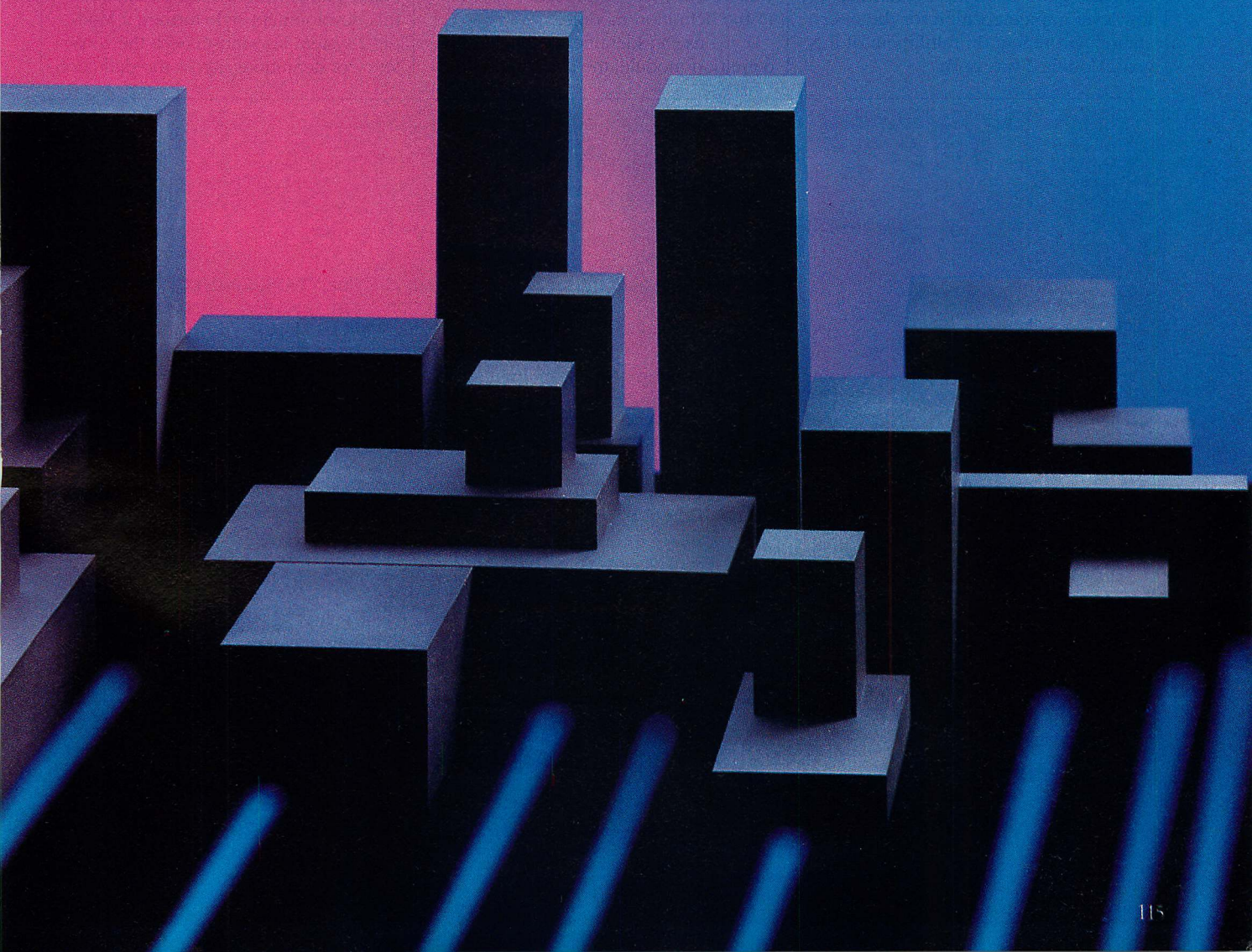
With PL360, Wirth initiated segmentation as a device for breaking a large program's code and data into smaller blocks. Noteworthy was his use of 360 condition code (the 360 equivalent of flags on the 8088). Also, in PL360, the Boolean expression governing an IF statement could be a simple test on conditional code. For example,

IF < THEN...

tested the condition code, which was set as a side effect of a previous arithmetic operation for being less than.

However, the language lacked a comprehensive set of I/O facilities and did not interface particularly well to OS, IBM's mainframe operating system. PL360 never fully took hold in IBM 360 mainframe assembly language programming and remained something of an oddity in the computer world.

In 1971, Wirth released Pascal, a language that gained widespread acceptance as a didactic tool and as a development medium on the PC, but Pascal still lacked multiprogramming capabilities. When Wirth began designing Mod-



PHOTOGRAPH • STEPHEN LONGLEY

ula, it was with the specific intent to implement multiprogramming, multiprocessing, and separate compilation.

In PL360 and Pascal, a program's source code was compiled all at once (using include files as necessary). As Pascal programmers know, this can protract development. With Modula, Wirth introduced separately compiled entities called *modules*. Modula's syntax also eliminates some of the earlier (more clumsy) language constructs, such as

```
IF ... THEN
  BEGIN ... END
ELSE
  BEGIN ... END
```

This particular sequence was supplanted with the simpler

```
IF ... THEN ...
ELSE ...
END
```

Modula was implemented experimentally in 1975. In 1977, Wirth began designing an integrated programmer's workstation, the Lilith. In developing Lilith software, Wirth's goal was to devise one language that would serve both at high-level application and at low-level implementation for the operating environment. In fulfillment of this goal, Modula-2 was born.

A LANGUAGE APART

In spite of its impressive heritage, Modula-2 has not yet achieved widespread acceptance within the PC programming community, a fact that is probably attributable to the instability of some of the early Modula compilers. Certainly the language is structured to answer the requirements of complex programming.

A complete Modula-2 program consists of one or more separately compiled modules, each of which has two parts, the *definition* and the *implementation*. The two usually are specified in separate files. The definition module contains the declarations of identifiers to be made visible to other modules; it resembles the header file in C, broken into small pieces. The names and arguments of procedure calls, data structures, and enumerated constants typically are written into the definition module. All of the systems reviewed here compile definition modules into a more compact form to allow the compiler faster access to symbol definitions. The details of how a particular module performs its task are hidden in an implementation module—the actual executable code for the procedure named in the definition module.

To use an identifier belonging to a definition module, the *client* module (the module desiring such use) must import it. The IMPORT statement specifies to the compiler the name of the definition module in which the identifier to be imported is defined. The compiler searches for and incorporates any symbols named in the IMPORT statement into the compilation of the client module. This includes procedure headers, along with definitions of procedure parameters and their types.

The philosophy behind the formulation of this construction is very powerful. In following such a format, project designers can specify the interfaces between procedures first, by building definition modules that define types and procedure headers. Then, varying implementations of a particular procedure can be substituted without ever changing the original definition, and because the implementation details are hidden from the other modules, no other module will come to rely on the details of implementation.

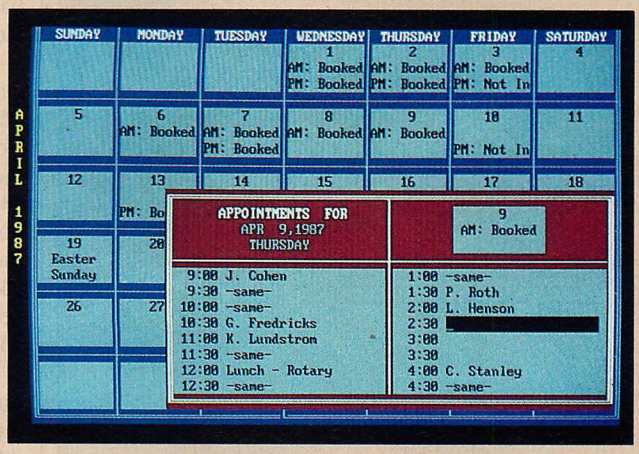
During compilation, only those definitions of interest to the client, as specified in an IMPORT statement, are included (unlike C, which includes the entire global header file). To enforce this development style further, a Modula-2 system also must check the consistency of definitions across the definition



BOB STANTON HAD A GREAT IDEA. AN HOUR LATER HE WAS TESTING IT.

Appointments. Everybody takes them — dentists, auto-body shops, dance instructors. And lots of computer applications need appointment screens.

Bob thought that a calendar made a terrific graphic metaphor for taking appointments. Simply use the arrow keys to pick an open date, then press the Enter key, and up pops an appointment window.



Lucky for Bob, he's a CLARION programmer, one of a fast growing cadre of super-productive application developers.

With CLARION's Screener utility, he painted a white calendar on a black background. Then he drew a white-on-blue track around the page and between the days. He typed in the days of the week — and *voila!* — a calendar!

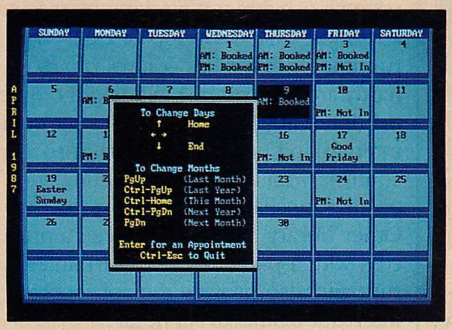
CLARION knows that a PC monitor is refreshed from memory, so it treats a screen layout like a group of variables. Just move data to a screen variable, and it shows up on the monitor.

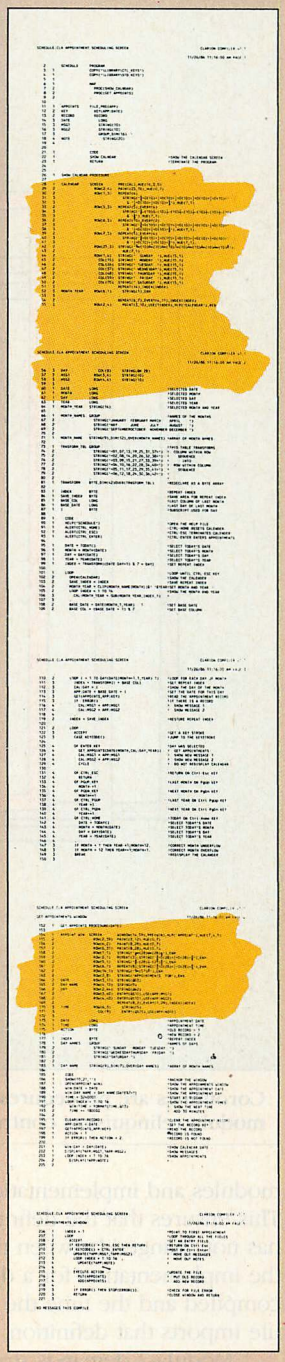
Bob set up dimensioned screen variables for the days of the month and a screen pointer for selecting a date, and he was done. Then Screener generated the code .

Then Bob drew the appointments window, built an appointment file, filled in the connecting code and tested it — ONE HOUR AFTER HE STARTED!

Testing was a breeze. Screener doesn't just write code, it compiles your source, displays a screen, gets the changes, then replaces the old code in your program.

So here are Bob's appointment screens. You can see the source listing to the right. We marked all the code Screener wrote for him.





WHY CLARION?

Why are application developers everywhere changing to CLARION?

Because CLARION gives you all the tools you need: a coupled compiler and editor; screen, report, and help generators; an import/export utility; a sort/backup/restore utility; a formatted file dump; a DOS shell — and much more.

Because with CLARION 's comprehensive data management routines, records can be locked and files shared on Novell®, 3COM®, IBM® PC Net & Token Ring, Multi Link®, and most other networks.

Because CLARION is *not* hardware locked or copy protected. Run-time systems are *free* and soon you will be able to translate CLARION into native machine code (.EXEs).

And best yet, the price of CLARION v1.1 is just $395 plus shipping and handling.

You'll need an IBM PC or true compatible with 320KB of memory and a hard disk drive. CLARION v1.1 also comes with a 30-day money back guarantee.

So call now and order CLARION v1.1. or ask for our detailed 16-page color brochure and reprints of major reviews.

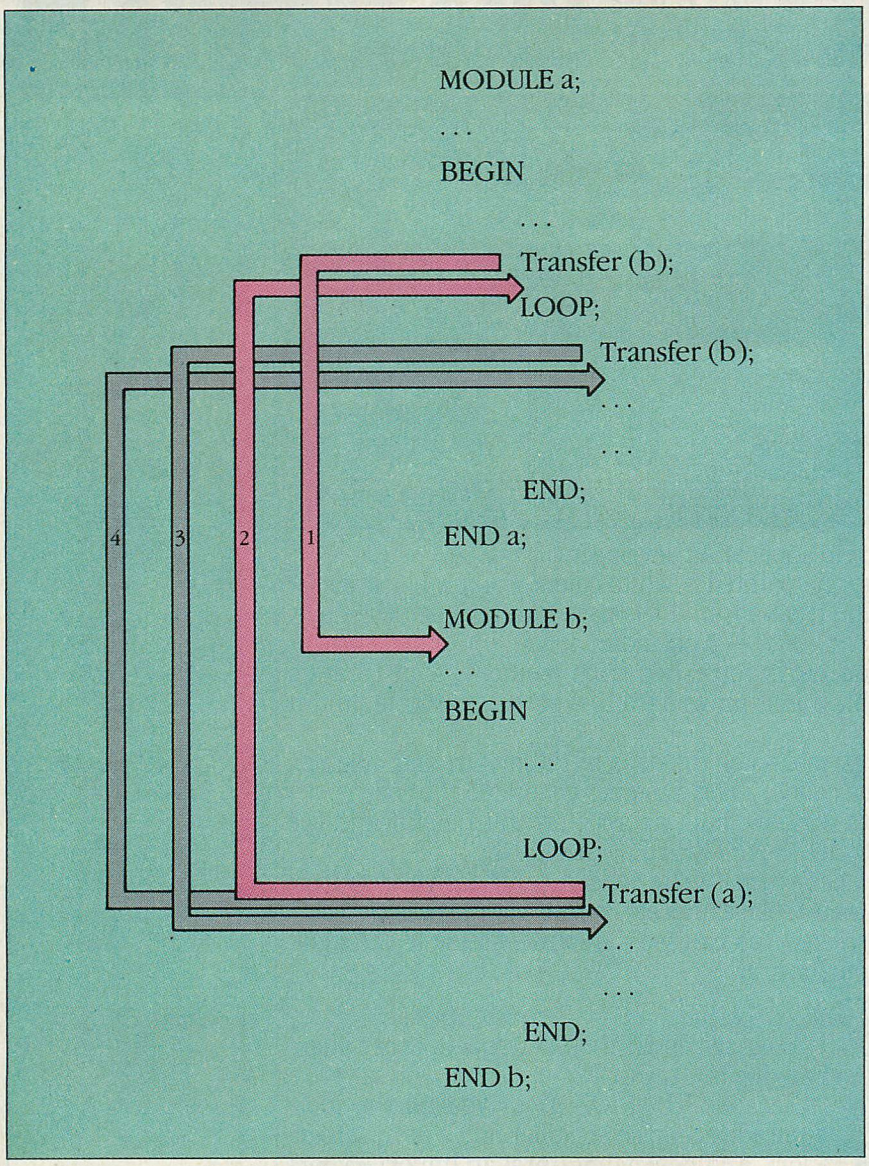
800/354-5444

CLARION®
from BARRINGTON SYSTEMS, INC.

150 EAST SAMPLE ROAD POMPANO BEACH, FLORIDA 33064 305/785-4555

Copyright 1986 Barrington Systems, Inc. CLARION is a registered trademark of Barrington Systems, Inc. IBM is a registered trademark of International Business Machines Corporation Novell is a registered trademark of Novell, Inc. 3COM is a registered trademark of 3COM Corporation Multi Link is a registered trademark of Software Link, Inc. Dept. A9

FIGURE 1: *Coroutine Execution*



Coroutines are procedures that share the same thread of execution. A Modula-2 module relinquishes control through a call to the procedure Transfer.

modules and implementation modules. This ensures that the definition module has not changed between the time that the implementation for a definition was compiled and the time the client module imports that definition.

Modula-2 demands strong typing of operands. The rules governing compatibility during assignment operations are strictly enforced, but may be defeated by explicit coercions, called *type transfers* (which are similar to casts in C and free union variant records in Pascal). Modula-2 library routines provide type conversions whereby the equivalent value is calculated. This differs from coercions in which the storage that is occupied by the identifier is reinterpreted in the coerced type.

Like C and Ada, Modula-2 has no built-in I/O constructs. In place of these, Wirth described a standard set of library modules containing procedures and functions (which in Modula return values) that perform basic I/O and floating-point arithmetic.

The language lacks a GOTO statement, a situation for which there are strong theoretical and practical reasons. It has been proved that any code sequence incorporating a GOTO can be implemented using structured logic (fully supported by Modula-2) with identical effect. Certainly, programs that contain GOTO statements are more difficult to debug and maintain. Eliminating GOTO also makes certain tasks easier on the compiler code generator.

Modula-2 permits absolute memory referencing in a standard, portable way; however, support for CPU register manipulation, address arithmetic, port I/O, and software interrupt invocation are handled differently by each of the Modula-2 implementations.

One of the language's more important capabilities is, of course, its support of multitasking and multiprocessing. However, the PC is a single CPU system; therefore, true multiprocessing, in which multiple threads execute simultaneously on separate CPUs operating in the same memory address space, cannot take place. As a result, Modula-2 processes on the PC are limited to quasi-concurrent serial execution.

Each of the systems reviewed here implements multitasking as a form of multiprocessing, and each offers two different sets of multitasking services. First, *coroutines* are procedures that share a single execution thread. This concept is presented graphically in figure 1. In that figure, MODULE **a** interrupts its flow of execution by performing a call to Transfer, to pass control of the CPU to MODULE **b**. MODULE **b** continues to execute until it calls Transfer, which returns the flow of control to the point in MODULE **a** at which execution was suspended by Transfer.

The second set of services—including a task creation mechanism and scheduling algorithm—delivers true multitasking. Most of these Modula-2 systems provide multitasking by placing a runtime system on top of DOS. The runtime system takes control of key hardware interrupts, including the timer tick interrupt 1CH, in order to set up a multitasking environment. Interprocess controls with appropriate waiting and posting mechanisms are provided. Other library routines associate a process with a particular hardware interrupt and also provide the appropriate interrupt dismissal services.

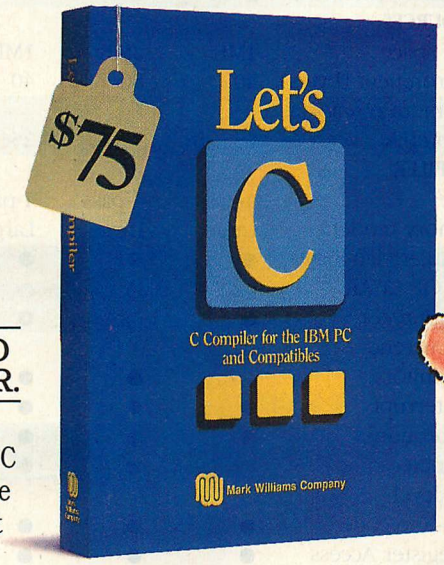
A program or implementation module that has a priority associated with its execution is called a *priority module* or *monitor*. The priority value is specified in square brackets after the name of the module in the implementation, as in the following example:

```
IMPLEMENTATION MODULE
    InterruptHandler [0]
```

The relative meaning of the priority number depends upon the Modula-2 implementation; in most cases, the lower the number, the higher the priority. When a procedure is called within a priority module, the current execution priority changes to the module's prior-

YOU ARE ABOUT TO BE SEDUCED BY POWER AND MONEY.

Admit it. You're intrigued with the idea of C programming. You may be working in BASIC, Pascal or Assembler now. But you're drawn to the power, portability and flexibility of C. And if money is what motivates you, imagine having it all for just $75 with Mark Williams Let's C.®



EVERYTHING YOU COULD ASK FOR IN A C COMPILER.

Let's C is no mere training tool. It's a complete, high quality C compiler. With the speed and code density to run your programs fast and lean. It won't get you sidetracked on some quirky aberration of C; Let's C supports the complete Kernighan & Ritchie C language–to the letter. And it comes from the family of Mark Williams C compilers, the name chosen by DEC, Intel, Wang and thousands of professional programmers.

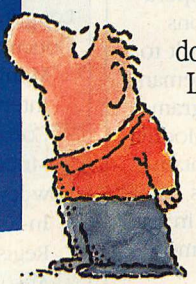
POWERFUL UTILITIES ARE A REAL BONUS

Let's C doesn't stop with being a high performance C compiler. It includes utilities you'd expect to pay extra for–like a linker and assembler plus the MicroEMACS full screen editor with source code included. Having the source code not only allows you to customize the editor, it offers a close up, fully commented view of C programming at its best.

REVIEWERS ARE SOLD ON LET'S C, TOO.

"Let's C is an inexpensive, high-quality programming package...with all the tools you will need to create applications."
–William G. Wong, *BYTE*, August 1986.

"Let's C is a thoroughly professional C environment loaded with tools and programming utilities...another fine Mark Williams product."
–Christopher Skelly, *COMPUTER LANGUAGE*, February 1986



"The performance and documentation of the $75 Let's C compiler rival those of C compilers for the PC currently being sold for $500... highly recommended..."
–Marty Franz, *PC TECH JOURNAL*, August 1986

Features

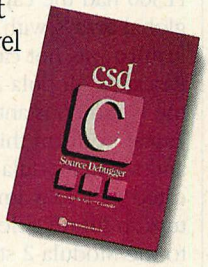
- For the IBM-PC and Compatibles
- Fast compact code plus register variables
- Full Kernighan & Ritchie C and extensions
- Full UNIX compatibility and complete libraries
- Small memory model
- Many powerful utilities including linker, assembler, archiver, cc one-step compiling, egrep, pr, tail, wc
- MicroEMACS full screen editor with source
- Supported by dozens of third party libraries
- Upgradeable to C Programming System for large scale applications development
- Not copy protected

Let's C Benchmark Done on an IBM-PC/XT, no 8087. Program: Floating Point from BYTE, August, 1983.

Exec Time in Seconds	
Let's C	134
MS 4.0	147

MARK WILLIAMS LET'S C
$75
60 DAY MONEY BACK GUARANTEE

ADD THE *csd* DEBUGGER AND CUT DEVELOPMENT TIME IN HALF.



Invest another $75 and you've got Mark Williams revolutionary source level debugger. *csd* lets you bypass clunky assembler and actually debug in C. That's a big help when you're learning C and indispensable when you're programming. *csd* combines the interactive advantages of an interpreter with the speed of a compiler, slicing development time in half. This is how Byte Magazine summed it up: "*csd* is close to the ideal debugging environment." William G. Wong, *BYTE*, August 1986

ARE YOU STILL RESISTING?

If there's any doubt that now's the time to get your hands on the power of C, consider Mark Williams 60-day money back guarantee. You can't lose. But with Let's C and *csd*, imagine what you could gain.

Ask for Let's C and *csd* at your software dealer's, in the software department of your favorite bookstore, through the Express Program at over 5500 Tandy stores or order now by calling **1-800-MWC-1700.***

*In Illinois call 312-472-6659

Mark Williams Company
1430 West Wrightwood, Chicago, Illinois 60614

© 1986, Mark Williams Company
Let's C is a registered trademark of the Mark Williams Company
UNIX is a trademark of Bell Labs.

MARK WILLIAMS LET'S C. ONLY $75.

CIRCLE NO. 207 ON READER SERVICE CARD

ity. A lower priority procedure can call a higher priority procedure; however, a runtime error occurs if a high-priority procedure calls one of lower priority. When a procedure of a given priority is running, only a procedure of higher priority may interrupt and run. The procedure Listen provides a mechanism to temporarily lower a module's priority in order to allow other modules of a lower priority to execute.

Modula-2 is wanting in some areas. Despite its claims to be a low-level language, for example, it places the BITSET type between the programmer and the bits that comprise a word. Instead of logical operations, such as OR, AND, and XOR, BITSETs use set operations to create logical bit operations. BITSETs themselves are not difficult to understand, but they do place a semantic layer between the system programmer and the hardware. Another Modula-2 weakness is that its sets can contain no more than 16 members. Types defined as SET OF CHAR, so useful in Pascal text processing, are illegal in most implementations of Modula-2.

As with standard Pascal, Modula-2 makes no provision for defining variables with an initial value. Although PL360 had the capability of predefining global arrays with constant data, this feature was not carried forward into Pascal or Modula-2. (Note, however, that the typed constants of Borland's Turbo Pascal perform this service very well, and some Modula-2 vendors have indicated they may implement a similar feature in future releases, as an extension to the Modula-2 standard.)

The Modula-2 language definition is presently quite spare because most of Modula's I/O and mathematical functions reside in libraries. Wirth described a library of basic routines in his report of the Lilith implementation (*Programmming in Modula-2*, third edition, Springer-Verlag, 1985). These descriptions were not meant as standards, and, indeed, are not rigorous enough to be considered standard definitions. Unfortunately, this situation has led to the development of dialects that hinder code portability; however, the British Standards Institute is close to releasing a report that does include definitions of basic library routines.

SYSTEM COMPONENTS

Since Modula-2 and its implementations were last reviewed, three new contenders have entered this compiler field, from ITC, PCollier Systems, and Workman & Associates. Pecan Software absorbed Volition Systems' p-code Modula-2, and enhanced it with a native code generator similar to that included with UCSD Pascal. Apart from the Volition Systems' metamorphosis, the systems reviewed two years ago are still going strong: Logitech has greatly enhanced its compiler and Modula Corporation has altered its system to produce native code. Table 1 lists the individual compiler specifications.

Most of these products are complex program development environments, with debuggers, editors, and utilities. In comparing the packages, several factors must be considered:

- The compiler itself must be bug-free.
- It must accept valid Modula-2 source code and reject invalid lines, and it must produce correct code for the source lines it compiles.
- The degree of completeness of each compiler's implementation of the language must be measured, including priority and process implementation, absolute variables, in-line low-level code, and interrupt access.
- Any nonstandard, but convenient, extensions that the compiler adds to Modula-2 that ease the programmer's burden must be assessed.

TABLE 1: *Compiler Specifications*

	INTERFACE	LOGITECH	MODULA	PCOLLIER	PECAN	WORKMAN
VERSION	2.0a	2.05	1.1	1.0	1.0	1.1
PRICE	$80.88	$89.00	$195.00	$99.95	$79.95	$49.95
FILTER/ENVIRONMENT	E	F	F	E	E	F
RESOURCES						
Disk space	1MB	2MB	1MB	1MB	750KB	500KB
Environment (bytes)	40	80	40	15	0	15
Minimum memory to compile	256KB	320KB	256KB	256KB	128KB	256KB
COMPILER						
Type	4-pass	4-pass	1-pass	1-pass	—[a]	1-pass
Memory model	Large	Large	Large	Small	—[a]	Small
Complete language	●	●[b]	●[b]	●	●	●
SET OF CHAR	○	○	○	○	○	●
Long integer	●	○	●	●	○	●
Concurrency						
Monitors	○	●	●	●	○	○
Interrupts	●	●	●	●	●	●
Coroutines	●	●	●	●	●	●
Multitasking	○	●	●	○	●	●
Low-level						
In-line code	●	●	●	●	●	○
Register Access	●	●	●	○	○	●
Absolute Variables	○	●	○	●	●	●
Interrupts	●	●	●	●	●	●
Real numbers						
In-line native 8087	●	●	○	○	○	○
8087/80287 library	●	●	●	●	●	●
Detect at runtime	○	●	○	○	○	●
Compiler options						
Storage squeeze	○	●	○	○	○	○
Runtime checks off	●	●	●	●	●	○
Defeat case	○	○	○	●	●	○
Listing	○	●	○	●	●	●
Syntax extensions						
Predefined arrays	○	○	○	○	○	○
Multiple while	○	○	○	●	○	○
EDITOR						
Syntax directed	●	○	—[c]	○	○	○
Syntax checker	●	●	—[c]	○	○	●
Multiple files	●	●	—[c]	●	●	●
Windows	●	●	—[c]	●	○	●

● = *Yes* ○ = *No*
Part. = *partial*
Opt. = *optional feature, at extra cost*

	INTERFACE	LOGITECH	MODULA	PCOLLIER	PECAN	WORKMAN
LINKER						
Output file extension	.EXE	.LOD	.EXE/.RLX	.COM	.CODE	.COM
Foreign object import	●	○	●	●	●	○
Overlays	●	●	●	●	○	●
Link time (seconds)	27	34	37	6	—[a]	62
Exec. size (bytes)	28,948	36,352	75,360	66,560	—[a]	19,456
RUNTIME SERVICES						
Dynamic linking	○	○	●	●	●	○
DOS spawn/chain	●	●	○	○	○	●
Command line	●	●	●	●	○	●
Get environ. strings	●	●	●	●	○	●
DOS error level	○	○	○	●	○	○
LIBRARIES						
Terminal I/O	●	●	●	●	●	●
Graphics						
Mice	○	●	●	○	○	○
Display adapters	●	○	●	○	○	○
File I/O	●	●	●	●	●	●
Random file I/O	●	●	●	●	●	●
Math (8087)	●[d]	Opt.	●	●	●	○
DOS services	●	●	●	●	○	●
String-handling	●	●	●	●	●	●
Sound	●	●	○	○	○	○
Comm port	●	●	●	○	●	○
Decimals	○	●	○	○	●	○
Source code	●	Part.	○	○	Part.	○
DEBUGGING SERVICES						
Runtime	●	Opt.	●	○	○	●
Postmortem	○	Opt.	○	○	○	○
Stack trace	○	●	○	●	○	○
TOOLS AND OPTIONS						
Source code	●	Opt.	○	○	○	○
Customize	●	Opt.	○	○	○	○
ROM package	○	Opt.	○	○	○	○
Language import	○	Opt.	○	○	●	○
Make	●	Opt.	○	○	○	●
Cross-referencer	○	Opt.	○	○	○	○
Precedence	●	Opt.	○	○	○	●

[a]Pecan Modula-2 runs only in the Power System, these items could not be determined or do not apply.
[b]See text.
[c]Editor not provided.
[d]Present, but operated incorrectly.

All of these compilers implement the de facto Modula-2 standard defined by Niklaus Wirth in his report of the implementation on the Lilith; some offer extensions.

- The limitations of the compiler must be considered—how many symbols are permitted and how maximum code block length are evaluated.
- The compiler's generated code should be of a high quality.
- The memory model for the compiled code must be evaluated.
- The compiler's overall friendliness, including diagnostic messages and graceful recovery from syntax errors, is also an important factor.

After assessing the compiler, the steps necessary to generate a Modula-2 program are examined by evaluating the mechanism for linking the modules together into an executable program. These systems generate a variety of executable programs: some are standard .EXE files, others are specific to the Modula-2 implementation and require the particular system's loader to run.

Each vendor's implementation of Wirth's standard Modula-2 libraries also must be examined: InOut includes procedures similar to C's streams; Terminal implements low-level terminal access routines; Storage contains memory-management routines; and MathLib0 provides floating-point procedures, such as trigonometric functions. In his report, Wirth also mentioned windowing routines for both text and graphics, graphic pointing device services, and a menu system. Most of these packages provide the basic routines Wirth described; some even provide the windowing interface used on the Lilith.

The development tools are considered last. The debuggers, program editors, and cross-reference utilities are evaluated for their usefulness and user friendliness. Any additional tools, cross compilers or ROM packages, for example, round out the review.

Code quality. Quantifying the "goodness" of compiled code is difficult. Code generation itself is a mathematically undecidable problem; hence a more heuristic approach is needed. The compiler author, when writing code generators, faces a trade-off between code speed and code size in mapping processor instructions to language constructs. In the 8086 instruction set, many routes may be available to generate code corresponding to a particular Modula-2 language statement. Code generators that use the 8086 architecture to the fullest generate the best code.

To write a code generator that produces good code, the compiler author must thoroughly understand the target machine for which the code is being generated and choose the code model that best fits the machine. A code model encompasses the procedure activation record (the return address, parameters, and temporary variables of a procedure), standardization of calling sequences, static data allocation and data representations—the framework into which the compiled instructions are placed. The appropriate choices here will vastly improve the speed of finished product: the emitted code. (Note that *code model* should not be confused with *memory model*, a term that defines pointer size and the use of code and data segments in allocating memory to code and data at execution time.)

The Intel/Microsoft standard code model requires building the procedure activation record by pushing the parameters onto the stack (the ordering is not important because Modula-2 does not support procedures with a variable number of arguments), and executing a call instruction, which places the return address on the stack. Open array arguments (unbounded parameter arrays) include their lengths as part of the parameter. The activation record pointer (in BP) is then created by pushing the old value of BP (the caller's activation record pointer) onto the stack and

FIGURE 2: *Code Quality*

```
MODULA-2 SOURCE CODE --

VAR i,j,k: CARDINAL;

j := 0;   k := 10000;
REPEAT
    k := k - 1;     j := j + 1;     i := (k * 3) DIV (j * 5);
UNTIL k = 0 ;


CODE GENERATION --
```

INTERFACE	LOGITECH	MODULA	PCOLLIER	PECAN	WORKMAN
mov j,0	mov j,0	mov j,0	mov j,0	mov j,0	mov j,0
mov k,10000	mov k,10000	mov k,10000	mov k,10000	mov k,10000	mov k,10000
mov ax,k	mov ax,k	mov ax,k	mov bx,k	mov ax,k	mov ax,k
sub ax,1	dec ax	sub ax,1	dec bx	dec ax	dec ax
mov k,ax	mov k,ax	mov k,ax	nop	mov k,ax	mov k,ax
mov ax,j	mov cx,j	mov ax,j	mov k,bx	mov ax,j	mov ax,j
add ax,1	inc cx	add ax,1	mov bx,j	inc ax	inc ax
mov j,ax	mov j,cx	mov j,ax	inc bx	mov j,ax	mov j,ax
mov ax,k	mov ax,k	mov ax,k	nop	mov ax,k	mov ax,j
mov cx,3	mov bx,3	mov bx,3	mov j,bx	mov bx,3	mov dx,5
mul cx	mul bx	mul bx	mov bx,3	imul bx	imul dx
mov cx,j	mov dx,j	xchg bx,ax	mov ax,k	mov bx,ax	mov dx,ax
mov bx,ax	push ax	mov ax,j	mul bx	mov ax,j	push dx
mov ax,cx	mov ax,dx	mov cx,5	push ax	mov si,5	mov ax,k
mov cx,5	mov si,5	mul cx	mov bx,5	imul si	mov dx,3
mul cx	mul si	xchg cx,ax	mov ax,j	mov si,ax	imul dx
mov cx,ax	mov dx,ax	xchg bx,ax	mul bx	mov ax,bx	pop dx
mov ax,bx	pop ax	mov dx,0	nop	cwd	call divide
xor dx,dx	mov cx,dx	div cx	mov bx,ax	idiv si	mov i,ax
div cx	mov dx,0	mov i,ax	pop ax	mov i,ax	
mov i,ax	div cx		xor dx,dx		
	mov i,ax		div bx		
			nop		
			mov bx,ax		
			mov i,bx		

The compilers perform relatively naive code generation. Notable inefficiencies are FTL's call to a function to perform division and PCollier's insertion of NOPs.

copying the current stack pointer into BP. Parameters lie at positive offsets from BP. Temporary variables are allocated on the stack below BP (that is, at a negative offset from BP) by subtracting from SP the number of bytes of temporary storage space required. At the end of the procedure, the temporary variables are removed from the stack by replacing SP with BP, in effect moving SP up past the temporary variables to BP. A RET instruction with a stack decrement value removes the arguments from the stack and returns control to the caller.

Among these compilers, the method for passing parameters to procedures changes depending on the code model. The Logitech, ITC, and Workman & Associates compilers all pass their parameters on the stack. By contrast, PCollier's compiler passes its parameters by pushing them onto the stack and, during the call, copying its parameters into the data segment of the target procedure. This method may permit faster accesses to the parameters as compared to finding them on the stack because the processor does not have to calculate an effective address for each, but it requires code to move the parameters and also to preserve them and any local variables in the event that the procedure is called recursively. Modula Corporation's compiler pushes its parameters onto the stack before calling the target procedure, leaving stack clean-up to the calling routine.

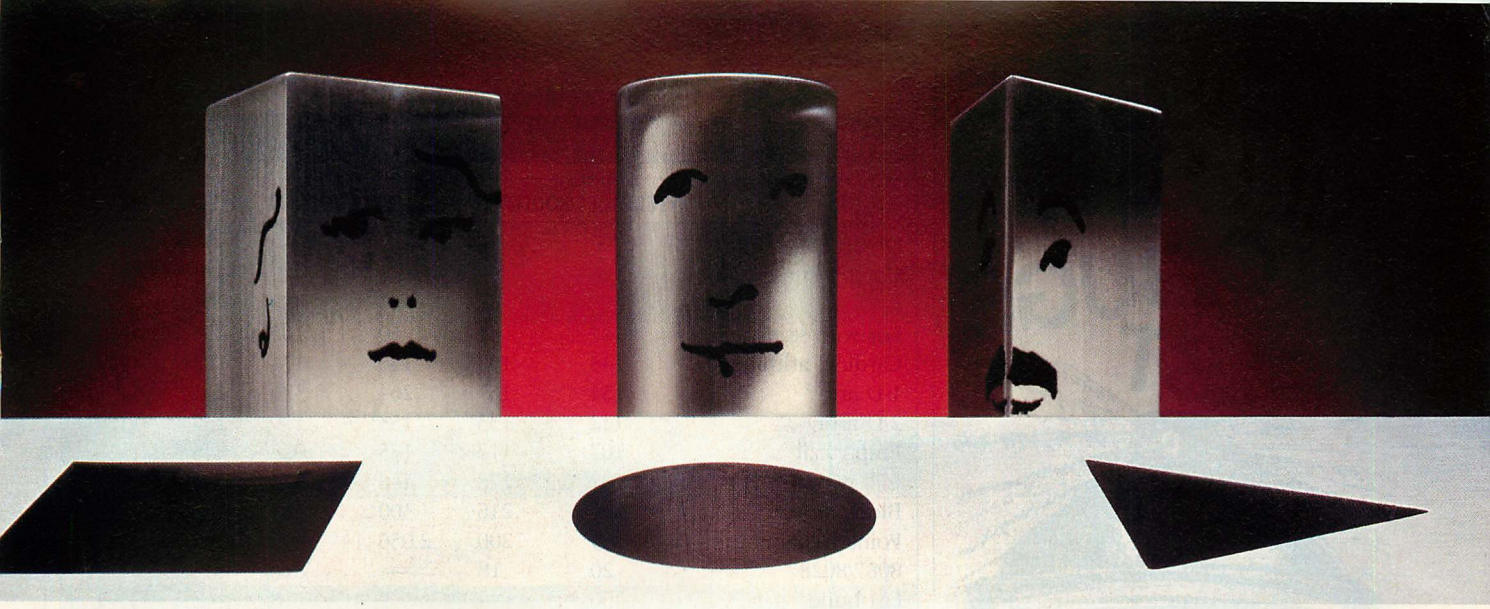
Apart from the choice of a method for calling procedures, other coding decisions must be made. The CASE statement, for example, can be compiled into several different sequences. Most compilers evaluate the case labels in the source code and watch for blocks of consecutive case labels. These blocks are then compiled into a jump table that points into the code generated for the CASE statement as a whole. The case label is translated into an index into the jump table. To cover nonsequential case labels, the code generated is typically a series of compares and jumps. Some compilers do not discriminate between consecutive and nonconsecutive case labels and simply generate a series of compares and jumps. Generating a jump table is possible during a one-pass compilation.

FOR loops also can be implemented in several different ways. The current Modula-2 definition specifies that FOR loops cannot have a variable step, and Wirth's Lilith report forbids changing the loop variable within the loop. This leaves the compiler author with several choices for generating the loop in machine code. For the most part, the compilers optimize on those loops with an increment of 1 by using the 8086's LOOP instruction. Upon entry into the loop, the compiler checks for degenerate cases (that is, it makes sure that the loop bounds are in the proper order and that the initial value for the loop index lies between them). Next, it calculates the number of times the loop must execute. The loop index and the loop count are then pushed onto the stack. The code of the loop is executed, and, at the bottom of the loop, the loop index and count are popped off of the stack and are updated. Then, the code branches to the top where the bounds are checked again, and so on, until the loop has executed the required number of times.

Very little optimization of code was observed among the reviewed compilers. In particular, the compilers were not sophisticated enough to "remember" that a useful register value had been loaded into a register for the use of a preceding statement. Instead, all necessary register values were reloaded each time a statement was translated to code, even when the proper values were present already.

Figure 2 provides a practical example of the differences in code generation. The figure shows a piece of Modula-2 source code and lists, side by side, disassembled listings of the cardinal code generated by each compiler (see table 2 for complete results from the performance benchmarks). Actually, the differences among the code samples generated were few: Modula Corporation's package performed the fastest because it did not use the stack for storing temporary variables; the FTL compiler pays a time penalty for calling a division subroutine; and PCollier's compiler has several NOP (no operation) instructions and insists on storing results through BX, as demonstrated by the last two instructions in its code sample.

The linkers. Each of the six compilers produces a proprietary object module format; therefore, each system must



The 3-in-1 applications development tool for every Tom, Dick and Harriett.



Presenting the first database applications development tool that delivers the promises of 4th-generation languages on the microcomputer.

It's SIMPLE Software.

SIMPLE stands for SYSTEM IMPLEMENTATION by EXAMPLE™ and it delivers the absolute best balance of power and ease-of-use.

SIMPLER THAN EVER BEFORE.

SIMPLE lets you jump in at any level of experience and unleash the full processing power of the PC. It has a totally non-procedural way of doing things that flattens-out the learning curve. You can automate everyday business tasks or prototype new applications without ever having to write a single line of procedural code.

SIMPLE enables experienced programmers to achieve higher levels of productivity. And for developing mainframe applications, SIMPLE is the perfect prototyping tool—combining ease-of-use, speed, and high-quality system documentation.

YOU GET 3-IN-1.

1. IT'S A DESIGN TOOL.

SIMPLE allows you to quickly prototype applications on the microcomputer—sit right down and draw, edit, and specify processing logic with a few keystrokes. SIMPLE aids in the process of structured design by organizing development efforts in a building-block fashion. Experienced developers move quickly and easily from task to task. Less experienced users are literally guided through the entire applications development process.

2. IT'S A RELATIONAL DATABASE MANAGER.

SIMPLE offers exceptional speed performance. It is written in Assembler and incorporates a highly optimized B-tree data access method that eliminates record sorting. SIMPLE also uses a dynamic single-record index. You can have an unlimited number of indices in any record. Each index may be either a single or concatenated key. SIMPLE's relational joins are easy to construct and more efficient to process. Joins are accomplished dynamically at time of processing and support one-to-many and many-to-many relationships.

3. IT'S AN APPLICATIONS GENERATOR.

It is the most practically functional applications generator on the market today. SIMPLE is completely non-procedural, never forcing you back to procedural methods, letting you accomplish even the most complex databased applications in the most easy-to-use-and-understand fashion.

A SINGLE, VISUAL 4GL LANGUAGE.

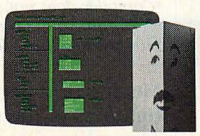
Uniquely, every step in program development from input screens, to reports, to complex processing logic is accomplished in SIMPLE's all-visual worksheets. It is fully non-procedural and totally picture-oriented. Design right on the screen, and SIMPLE's built-in pattern-recognition logic automatically generates all the application code for you. You never have to leave SIMPLE's 4GL non-procedural technique.

POWERFUL WORKSHEETS.

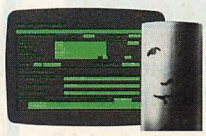
SIMPLE provides three powerful, yet easy-to-learn worksheets—a File Worksheet, Specify Worksheet and Design Worksheet. You need only to define a file, then SIMPLE creates the basic program for you. Enhancements are done on SIMPLE's Design and Specify Worksheets.

In the Design Worksheet, you simply paint or draw an example of the input screen or report you want. The Design screen may be painted exactly as you want using SIMPLE's built-in, full-screen editor which offers a wide range of capabilities to aid you—including the ability to delete or insert a character or an entire line, move or copy blocks of information, lasso text or variables, and window to other worksheets in one or two keystrokes. Powerful specification macros are invoked providing application users the ability to pop-up a window and browse through another file, interrupt data entry to perform another program, provide context-sensitive help, and perform conditional processing based on the user's input.

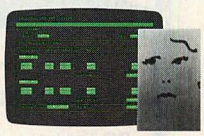
In the Specify Worksheet, you implement your processing logic. You never have to fall back to procedural programming to get the processing power you need. Range checking and data validation are easily implemented, visually. Conditional processing statements are quickly set up. Arithmetic operators, date operators, and a full set of string functions are available. You specify an example of how you want your data processed, and SIMPLE creates the program.



Company Managers like Tom **easily develop customized applications like this Branch Reporting System which reports information from support service calls. Tom now has a system which validates certain information and provides a customer history to improve the branch's support capabilities. With SIMPLE's Specify Worksheet on screen, Tom simply joins data from four different files and establishes their relationship.**



Information Center Staff Members like Dick **easily create new microcomputer applications systems like this Lead-Tracking System. Dick is able to sit down with the Director of Marketing and review the main entry screen developed on SIMPLE which shows the prospect demographic information, the media source and date from which each lead was generated, and the fulfillment literature to be sent.**



System Analysts/Programmers like Harriett **easily prototype design changes while interacting with Department Heads. Harriett has prototyped some requested changes in a Mainframe Payables System. With the help of SIMPLE, she has built a test database with data imported from the mainframe and is able to review the check-ledger report, in the Controller's office on her portable computer.**

A POWERFUL MENU GENERATOR.

SIMPLE's power gives you the easy flexibility to generate unlimited levels of user-guidance menus, as your applications require. There are limitless design options for effective "point-and-shoot" menu creation. And you have total freedom to build menus before, during or after development, unmatched in other systems.

SIMPLE, SIMPLER, SIMPLEST.

Ask for SIMPLE at your computer dealer. Or call us direct for the dealer nearest you and a full-functioning SIMPLE demo package with a Quick-Start manual for only $9.95*.

*Includes shipping and handling. California residents add sales tax.

SYSTEM IMPLEMENTATION BY EXAMPLE is a registered trademark of Accuphase, Ltd.
IBM PC is a trademark of International Business Machines, Corp.

SIMPLE

SYSTEM IMPLEMENTATION BY EXAMPLE™

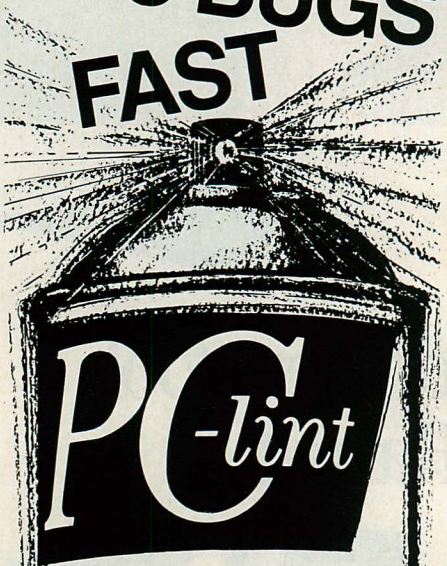
Published by Software Merchants Unlimited

Software Merchants Unlimited
2252 Fillmore Street, Suite 401
San Francisco, California 94115
415-567-5071

CALL 800-8 SIMPLE

#1 Lint for MS-DOS

KILLS C BUGS FAST



PC-lint

The professional diagnostic facility for C

PC-lint lets you zap swarms of C bugs and glitches at a time.

Now you can uncover the quirks, inconsistencies, and subtle errors that infest your C programs . . . waiting to bite you. PC-lint finds them all . . . or as many as you want . . . in one pass. Set PC-lint to match your own style.

Outperforms any lint at any price

- Full K&R support and common ANSI enhancements (even MS keywords)
- Finds inconsistencies (especially in function calls across multiple modules!)
- Modifiable library descriptions for 8 popular compilers
- Super fast, one-pass operation
- Suppress any error message
- Zillions of options

PRICE $139 • MC • VISA • COD

Includes USA shipping and handling. Outside USA, add $15. In PA add 6%.

ORDER TODAY, 30-day guarantee

Runs under MS-DOS 2.0 and up, and AmigaDOS. Uses all available memory.

Trademarks: PC-lint (Gimpel Software). MS, MS-DOS (Microsoft), Amiga (Commodore)

GIMPEL SOFTWARE

3207 Hogarth Lane, Collegeville, PA 19426

(215) 584-4261

TABLE 2: *Performance Benchmarks*

	INTERFACE	LOGITECH	MODULA	PCOLLIER	PECAN	WORKMAN
Object size (bytes)	3,196	4,092	5,830	6,760	—[a]	8,192
Compile time	26	58	45	42	160[b]	13
Repeat	53	49	49	88	45	46
For with step	47	41	59	106	57	52
Cardinal arithmetic	148	150	145	171	83	173
1-D array	154	167	264	326	261	145
2-D array	122	133	153	179	137	125
Empty call	107	112	175	928	545	68
Call with 4	182	178	446	1,065	604	153
Block move	217	216	300	304	145	213
Pointer chaining	269	200	2,056	498	130	143
8087/80287	20	18	—[c]	816	—[c]	—[d]
List build	456	4	7	8	—[e]	1
List dispose	460	600	30	5	—[e]	3
Eratosthenes Sieve	12	14	20	32	18	12
WriteString	140	126	146	166	253	224

All times are in seconds.
[a]Object size could not be determined; see text.
[b]These numbers are the result of dividing the actual times taken by 60. This compiler's Time function was undocumented—the resolution of the clock is in 60ths of a second.
[c]Compiler does not support the 8087/80287.
[d]A runtime bug occurred, preventing completion of the test.
[e]The runtime heap was too small to complete this test.

Much of the performance difference among compilers is related to the quality and tightness of the runtime support libraries and the efficiency of procedure calls.

provide its own linkers in order to produce an executable file. A linker is also desirable for other reasons. In a Modula-2 environment, the linker should be able to search for the component modules of a program as they are specified in the main module, without having to name them at the invocation of the linker. DOS LINK cannot do this.

Among these compilers, two approaches are used to link the executable file: *static* and *dynamic*. Static linking is the method in which a linker adjusts address references in a relocatable object module and optionally combines the module with other modules to produce a single executable code file. In dynamic linking, portions of the executable image can be retained on disk and loaded into memory as required and as memory becomes available. Dynamic linking is similar to having procedures linked as overlays except that overlay procedures are linked for a specific code segment offset. Routines loaded by a dynamic linker can be placed anywhere, and segment references are fixed only when the code is loaded. To understand dynamic linking, it is necessary to know how object modules are put together into executable programs that the operating system can run.

Static linking builds an executable code file. This file contains the executable machine instructions as well as relocation information for DOS; thus, instructions that use segment addresses can be adjusted to point to their absolute locations in memory. DOS fixes those instructions before starting execution. If the program implements overlays, the overlay manager calls DOS to load overlays as required, supplying DOS with the paragraph address for segment address fix within the overlay.

In dynamic linking (as used in Microsoft Windows), the executable image contains pointers to the names of routines that are to be linked when called at runtime. Routines to be dynamically linked are specified in the executable file. All static links are fixed as the executable file is processed by the linker. At runtime, the dynamic linking manager finds the requested routines and permits callers to call them. The fundamental difference in dynamic linking is that the routines are referenced by name, so that they can move around in memory from one call to the next.

Windows implements this feature as tables of *thunks*—snippets of code in a fixed location that always point to the entry of a routine, or to a fault manager that has the responsibility to reload the linked routine. The dynamic links are resolved at runtime.

Some of the implementations reviewed provide *dynamic loading* as a means of quickly checking out a pro-

gram during development. In dynamic loading, linkable object modules are loaded from disk at execution time, with all fixes occurring while the object modules are being loaded. Dynamic linking, by contrast, is done partly by the linker at link time, when intrasegment references are fixed. Intersegment references, however, are not fixed until execution time, because until that time, the runtime system does not know the location of any given segment.

In dynamic loading, a loader executes with the main module of the program. Procedure calls are intercepted by a runtime monitor that loads the appropriate module. As execution proceeds, modules that are called are loaded in turn. Dynamic loading is very handy during development, when a system requires a half hour to link statically. However, dynamic loading takes a toll on performance, as evidenced by the PCollier subroutine call benchmark. When producing a final product, the dynamic links need to be resolved to eliminate the performance penalty.

Most of the static linkers in these packages fix the instructions directly where the intermodule references are made. The Logitech and ITC compilers generate a FAR instruction, either a CALL or a JMP, leaving four bytes in the instruction's fields for the destination address to be filled in by the linker. Workman's FTL uses the small memory model, so only NEAR references are used, requiring only two bytes of space for the fix. Modula Corporation resolves references to a table of addresses at the beginning of each code segment, so indirect CALL and JMP instructions are made through this table.

PCollier's static linker does not fix the references per se, but instead performs a permanent dynamic link. The compiler and the loader assign a numeric identifier to every procedure in the program. When a subroutine call needs to be made, the caller loads the identifier of the target procedure into a register and calls a module manager that deciphers the identifier and passes control to the selected routine. As born out by the benchmarks, this method is slow.

MEASURING MODULAS

Each compiler was programmed with updated versions of the Modula-2 benchmark programs written for the previously mentioned articles. (The source code for the benchmarks as they execute under the compilers tested is available on PCTECHline). Here the speed of the compiled code was tested in six specific areas: execution of several empty loops, integer arithmetic, array indexing, procedure calls, block moves, and pointer chaining. The classic Sieve of Eratosthenes was included as a second test of integer arithmetic. The speed of each system's libraries was tested in three ways: calculating a real number, writing to the screen, and creating and disposing of a linked list. (In order to put all of the compilers on equal footing, runtime checking, if available, was turned off.)

Porting the benchmark to each system required minor adjustments to the programs. Each language system has a different interface to the system clock, requiring a modification to the code. In all cases, this was a trivial change.

The compilation speed of each compiler was measured as it compiled its benchmark program. Most measurements were timed by the system clock using batch files. Because the nature of their environments precluded using batch files, the ITC and Pecan products were timed using a stopwatch. The test machine was a standard IBM PC with 640KB memory, a 20MB hard disk, and DOS 3.1 configured with 20 buffers. The actual compiled code was examined with Microsoft SYMDEB by tracing through the various Modula-2 runtime systems to get to the benchmark pro-

PC ⇔ MAINFRAME
VIA 9-TRACK TAPE

For Information Interchange – Backup – Archival Storage

IBM format compatible 9-track, ½ inch magnetic tape is the universally accepted media for mainframes and minicomputers. Catamount offers *Low Cost, Lightweight* 9-track Tape Subsystems for the IBM-PC/XT/AT computers which allow:

- Reading tapes generated on mainframes and minicomputers.
- Writing tapes to be read on mainframes and minis.
- ASCII, EBCDIC and Binary tapes accommodated.
- 800 bpi NRZI, 1600/3200 bpi PE, and 6250 bpi GCR format systems available.
- Storage capacities up to 270 MB on a single reel.

Systems come complete with comprehensive DOS command syntax oriented software and an Installable Device Driver. For OEM applications, the tape controller is available separately.



CATAMOUNT CORPORATION
2243 Agate Court
Simi Valley, CA 93065-1898
(805) 584-2233
FAX: 805-584-0941

CIRCLE NO. 111 ON READER SERVICE CARD

VT100/VT52 & Tektronix™
4010/4014 Terminal Emulator
Excellent emulation and the features you want:

VTEK™ 3.1

- use 4096 x 3120 resolution
- zoom, pan, and window plots
- high resolution printer dumps
- choose text and plot color
- transfer files with XMODEM and Kermit protocols
- scroll last 4 pages of text
- 132 column VT100 capability
- 18 User-definable keys
- capture plots and text on disk
- full or half duplex
- access to DOS commands
- all VT100 keypad commands
- command line editing
- fast direct screen access
- password security

VTEK makes your PC better than a terminal
$150 from Scientific Endeavors

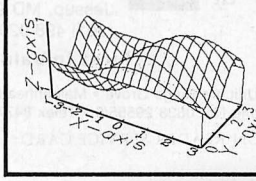
Publication Quality Graphics for
Scientific and Technical Applications

GraphiC™ 3.0 in Color

- linear, log, & polar plots
- bar charts & Smith charts
- contour plots with labels
- 3-D curves, 3-D surfaces with hidden line removal
- 4 curve types, 8 markers
- 14 fonts, font editor
- multiple levels of super/sub scripts
- 4096 x 3120 resolution
- zoom, pan, window plots
- multiple plots on a page
- high resolution printer dumps, full or half page
- plotter support in COLOR

16 color plots on EGA, Sigma, TeleVideo & Tecmar boards
Over 100 routines can be called by your C program. $350. Demo $8.
SOURCE INCLUDED for private use only.
For DeSmet, C-86, Aztec, Lattice, and Microsoft C compilers.

Scientific Endeavors
Route 4, Box 79; Kingston, TN 37763
(615) 376-4146



For 256k IBM and Corona PCs, DOS 2.xx,3.xx. Epson, Okidata, Toshiba, C. Itoh printers. Hewlett Packard, Houston, Sweet-P plotters. Corona Laser printer. IBM, IBM EGA, Sigma, TeleVideo, Tecmar, Hercules, Corona graphics. A compatible assembler is required.
THIS AD WAS MADE USING GraphiC™

CIRCLE NO. 187 ON READER SERVICE CARD

gram. Pecan's generated code was not traced by SYMDEB due to the presence of p-code in certain library routines; instead, it was disassembled to get an idea of its quality by visual inspection.

For the most part, the compilers performed comparably in the benchmarks. Subtle differences in the code sequences for the benchmarks explain some of the variation, especially in the empty loop test, FOR loop tests, and cardinal arithmetic. In evaluating the cardinal arithmetic and the Sieve benchmarks, the differences in the timings reflect the differences in the code that is generated (refer to figure 2.)

The indexing tests show ITC to be a winner—it makes the best use of the 8086 index registers. The differences between ITC, Logitech, and Workman in this test are due to ITC's improved register deployment and arithmetic expression generation. Logitech reloads its segment registers needlessly during this test, slowing the code down.

In evaluating the subroutine call benchmarks, the differences between ITC and Logitech are slight. Workman's good score reflects the small memory model's single code segment and its 16-bit return addresses, which use less time in transferring control. Modula Corporation does not use the stack as effectively; it dissolves the stack "manually" using five instructions after the CALL, rather than letting the CPU perform the task with one instruction. PCollier, as stated above, uses a dynamic linking mechanism, and copies its parameters from the stack into the called routine's data segment.

In performing a block move when assigning one structure to another, the fastest performers use a REP MOVSW instruction sequence. The PCollier and Modula Corporation compilers use MOVSB, resulting in times that were 50 percent longer. When the results of the pointer chain test are analyzed, they result in a tie between Logitech and Workman's FTL, because FTL uses 16-bit pointers (which are passed around more efficiently). Modula Corporation's compiler calls a runtime subroutine to perform the chaining; moreover, this product implements pointers backward from the Intel standard, which means that the compiler cannot use the 8086's pointer instructions LDS and LES.

Logitech and ITC also did quite well in the floating-point test because they both offer in-line 8087 code generation. Modula Corporation's entry was disqualified when it claimed a spurious floating-point error.

Apart from the Workman FTL, which in the small memory model uses only 16-bit pointers, those compilers adhering to the Intel code model performed the best: Logitech and ITC.

The libraries were tested only briefly. Here the variation in performance is due entirely to library efficiency, affected to some degree by the efficiency of the compiler. The wide variation in performance of the heap management routines (list build and list dispose) is curious. Without the library source code (provided only by ITC), it is difficult to evaluate further.

Interface Technologies Corporation. Ambitious as it was (and in spite of the fact that many bugs present in earlier versions have been fixed), the ITC compiler narrowly misses the mark. ITC markets two versions of its Modula-2 system, the Modula-2 Software Development System, or M2SDS, and SDS-XP. (See the review of M2SDS in Product Watch, Cole Brecheen and Charles Bradford, September 1986, p. 187.) Both products contain the same editor/compiler; the XP version includes additional tools and source code files.



The Digi-Data 2000 PC tape system reads and writes IBM/ANSI compatible, 9 track, 1600 bpi, ½ inch tapes. It comes complete with PC controller board, cables and DOS software utilities. Just plug it in and run.

The 2000 PC provides file interchange in ASCII, EBCDIC or binary. That means you can exchange data between your PC and most minis or mainframes.

The 2000 PC also provides high speed disk backup and restore functions.

For all the reasons you need a 9 track tape on your IBM PC/XT/AT, call us at (301) 498-0200.

DIGI-DATA CORPORATION
8580 Dorsey Run Road
Jessup, MD 20794-9990
(301) 498-0200 Telex 87-580

... First In Value

In Europe contact: Digi-Data Ltd. • Unit 4 • Kings Grove • Maidenhead, Berkshire England SL6 4DP • Telephone No. 0628 29555/6 • Telex 847720

CIRCLE NO. 123 ON READER SERVICE CARD

ITC offers a rich development environment with a windowing syntax-directed editor/compiler. The editor is the first pass of the compiler. Alt-key combinations are used to insert syntactic skeletons, then blanks are filled in. The editor "knows" the syntax of the language: inappropriate statements cannot be entered, meaning that syntax is checked, in effect, during program entry. The editor also controls the formatting of text lines: the user cannot, for example, place two statements on a single line. Program source code is stored in a compressed form, thus speeding the compilation process.

As radical an advance as this editor represents, it is a frustrating tool to learn—knowing where and when to insert a syntactic skeleton takes time to learn. All previous typing and program editing instincts must be discarded.

The editor supports multiple, overlapping windows, and text transfer between windows. Code generation options (for example, in-line 8087 code and runtime error checking) are set via a Ctrl-key combination, and remain in force only for the duration of the development session within a window. To generate object code, the user enters another Ctrl-key sequence and the compiler compiles the program source code to object code. If an error is discovered at compile time, the editor is positioned at the offending line.

Following a successful compilation, the user exits the editing window and calls up the linker, choosing the main program module just compiled from a menu of all compiled modules in the current library. The linker then produces a .EXE file and a symbol map in the user's DOS directory. The .EXE file is fairly straightforward to follow with SYMDEB. Care must be taken, however, when tracing around INT E4H instructions, which ITC uses to implement calls for runtime services. Because the parameters for the calls are placed as data in-line below the INT instruction, the user must not set a breakpoint directly below the INT. Control does not return to the location immediately following the INT, and the debugger's breakpoint instruction will be confused with parameter data.

This compiler is nearly complete in its implementation of Modula-2, up to the second edition of Wirth's book. Monitors cannot be created—this editor simply does not permit the syntax. Definition modules must explicitly export their identifiers. Absolute variables are not allowed, again because the editor does not support the syntax. The compiler recognizes strings as a special type of variable, implemented in Turbo Pascal fashion, by specifying a maximum physical length in the declaration. String logical length is maintained in the first byte of the array, and changes as the string is manipulated.

The code model for the ITC compiler follows the Intel/Microsoft standard model. Parameters are passed on the stack, function procedure return values are placed in registers, and the called procedure cleans up the stack on exit. The ITC benchmark results bear this out—this compiler's performance is clearly in line with Logitech's.

For low-level issues, access to 8086 registers is permitted by importing register variables from the SYSTEM module. The register variables can be included in assignment statements, rather than being accessed as record fields or via procedure calls. SYSTEM also provides a set of routines to access the hardware interrupt vectors. Interrupt service routines written in Modula-2 can be attached to hardware interrupt vectors. In-line assembly language code is

Lattice® Works

LATTICE ANNOUNCES MICROSOFT WINDOWS SUPPORT IN VERSION 3.2

Version 3.2 of the Lattice MS-DOS C Compiler features full support for Microsoft Windows—including the "far," "near," and "pascal" keywords.

In addition, version 3.2 includes the ability to generate more than 64K bytes of static data and to declare objects larger than 64K bytes. It also includes improved support for ROM-based applications via the "const" data type. Version 3.2 is a significant release because it eliminates Microsoft's claimed monopoly on future MS-DOS C development tools. Now that the Lattice MS-DOS C Compiler supports a window interface, programmers using Lattice C can avoid the problems caused by switching to a different compiler. $500.00

LATTICE NOW OFFERS ENHANCED AmigaDOS C COMPILER

Version 3.1 of the Lattice AmigaDOS C Compiler offers a new library with 100 more functions than the standard AmigaDOS C Compiler. What's more, increased library modularity and new addressing modes help reduce load module sizes by more than 20%. The new version also features faster pointer and integer math, faster IEEE floating point routines, direct support of the Amiga's FFP format floating point library, and multi-tasking support.

With Version 3.1, Lattice has broken free of the reliance on the Amiga standard linker and object file format. This new release includes completely new expanded documentation, and a Lattice assembler and linker which remain compatible with previous software but allows professional programmers to take advantage of both the Amiga's speed and the industry's standardization.

Lattice AmigaDOS C Compiler with Lattice's Text Management Utilities, $225. Professional AmigaDOS C Compiler with, Text Management Utilities, Lattice Make Utility, Lattice Screen Editor, and the Metadigm MetaScope Debugger, $375. AmigaDOS C Compiler $150.

LATTICE RELEASES NEW VERSIONS OF C CROSS COMPILER AND LINKER

Version 3.1 of the Lattice C Cross Compiler to MS-DOS and version 2.12 of the Plink86Plus Overlay Linker are now available for Sun and Apollo workstations as well as the DEC VAX Family of processors running VMS, UNIX or Berkeley UNIX.

All Lattice C Cross Compilers possess the same functionality and generate the same code as the native Lattice MS-DOS C Compiler. This allows users to take advantage of the larger systems' speed and multi-user capabilities when creating applications for most popular PCs.

Contact Lattice Corporate Sales for details.

Lattice

(800)533-3577 In Illinois (312) 858-7950 TELEX 532253 FAX (312) 858-8473

INTERNATIONAL SALES OFFICES: Benelux: Ines Datacom (32)2-720-51-61
Japan: Lifeboat, Inc. (03)293-4711 England: Roundhill (0672)54675
France: Echosoft (1)4824.54.04 Germany: Pfotenhaur (49)7841/5058
Hong Kong: Prima 85258442525 A.I. Soft Korea, Inc. (02)7836372
Australia: FMS (03) 699-9899 Italy: Lifeboat Associates Italia (02) 46.46.01

CIRCLE NO. 160 ON READER SERVICE CARD

supported through the pseudo-procedure CODE, exported by SYSTEM. Processes are implemented only as coroutines under ITC Modula-2. No provisions are made for multitasking.

The extensive libraries include Wirth's standard set. Some extra libraries provide screen graphics, including support for region clipping and coordinate transformation, management of the machine's serial communications ports, and ITC's unique file system. The performance of Storage is slow, but it is about equal to Logitech's version.

The XP version includes a make utility and a foreign object code import facility. The latter facility is limited to one code segment; it ignores FIXUPP object records. (FIXUPP records indicate operations emitted by the assembler's instructing the linker to place an address at the designated location. For more about object record formats, see ".OBJ Lessons," Steven Armbrust and Ted Forgeron, October 1985, p. 62.) This limitation renders the facility virtually useless because assembled instructions within an imported segment cannot reference locations within the segment, such as in-line tables.

Additional features to this package are a calculator, an ASCII table, a clock, and a file display utility. The system also has provisions for exporting and importing source code modules to and from DOS as ASCII files.

The system's documentation is good, although it is a bit unorganized. However, the tutorial in the beginning of the manual is a fine introduction to Modula-2 programming.

The ITC system incorporates some novel program development ideas. The editor was one of the first syntax-directed editors used for development on the PC, and it is very efficient. The compiler itself is reasonably fast and compact. Yet, with all it has going for it, this system has problems. The editor hangs occasionally, without warning or apparent reason, and the compiler sometimes generates incorrect code. For example, the statement

```
Boolean := (int > 0);
```

generates erroneous code. Moreover, the compiler yields an "Expression too complicated errors" message for the most trivial of expressions, such as in

```
d := a*b + c*d;
```

which generates an error when the four variables are defined as LONGINTs.

FOR loops with variable steps, illegal in Modula-2, are accepted by the compiler and then generate bad code. Changing the FOR loop index variable is not ignored. (All of the other compilers generate code sequences that preserve the loop variable and restore it with each iteration of the loop.) For example, for the statement

```
FOR i := 1 TO 10 BY k DO
  j := j + 1;
END;
```

the compiler accepts this code, which is itself an error, because no FOR loop can have a variable step. (The k was previously declared an integer variable.) The disassembled code is as follows:

```
top: cmp  i,10          ;check loop bounds
     ja   out
     mov  ds, cs:[0]
;get the module's data segment
     mov  ax,j
     add  ax,1
     mov  j,ax          ;j := j + 1
     add  i,4
;ERROR! '4' was pulled out of the air
     jmp  top
```

In the line where the step value is added to the index value i, the code

REQUIRED READING FOR ALL IBM PC SYSTEMS EXPERTS

If you're a systems expert, microcomputer specialist, or MIS/DP professional working with IBM PCs, you need the comprehensive information PC TECH JOURNAL provides 13 times a year!

It's the only magazine that provides you with the technical information to help you increase the performance of your multi-component system. It talks to systems experts and systems designers in the language you understand—about the applications and products you have to know about!

As part of your annual subscription to PC TECH JOURNAL, you'll receive the special PC TECH JOURNAL Directory issue published in November, the most comprehensive guide and index to the products in the PC marketplace and PC TECH JOURNAL's coverage!

Don't leave a gap in your required reading, subscribe to PC TECH JOURNAL now and *save 50%!*

For faster service call Toll-Free 1-800-852-5200 today!

TECH JOURNAL *P.O. Box 2966—Boulder, CO 80322*

SAVE 50%!

Send me PC TECH JOURNAL for:

☐ One year (13 issues) for $26.70.

☐ Two years for $53.35.

Savings based on annual single-copy price of $53.35.

Mr./Mrs./Ms. _____
(please print full name)

Company _____

Address _____

City _____ State _____ Zip _____

☐ Bill me ☐ Payment enclosed

Add $6 per year for postage outside USA, US currency only. Please allow up to 60 days for delivery of first issue. Annual Basic subscription price is $34.97.

4Z650

adds constant value 4 to it. The 4 was not specified in any way in connection with the loop, and it is unclear how the compiler decided to use it. If it had chosen a 0 instead, the code would have gone into an endless loop.

In spite of these seemingly major problems, the M2SDS package is selling well in Europe, where it was not available in its earlier, more flawed, releases. Perhaps in time the product will be improved and make a comeback in the United States as well.

Logitech, Inc. Logitech's Modula-2 appears to be the clear leader in the Modula-2 compiler field (in spite of the fact that it, too, had a number of bugs in its early releases). Also, before its several components were unbundled, Logitech's compiler was an expensive tool for much of the user audience. (See the review of Logitech's "Modula-2/86" in Product Watch, John T. Cockerham, September 1986, p. 187.)

The Logitech system consists of a compiler, a windowing editor, a linker, and a .EXE file creation utility. The compiler is available in two forms: overlaid and fully linked. The overlaid compiler runs in 256KB, and is available with or without 8087/80287 numeric coprocessor support. The fully linked compiler requires 512KB of RAM, but allows more symbols and code space and includes support for an 8087/287 and the 80186/286, a make utility, a windowing environment, and a source-level runtime debugger. Other optional features to the product are a text windowing library, a ROM package, and a Turbo-Pascal-to-Modula-2 translator.

The Logitech editor is an impressive one, supporting multiple windows and files. This editor has some awareness of Modula-2 syntax; it knows, for example, that it must indent after certain statements have been entered onto a line. In addition, a syntax checker that can be invoked from the editor via a function key reports lapses in indentation style. Both the compiler and linker are invoked via a function key as well, and the compiler passes error location information back to the editor for positioning of the cursor.

This system's source code files compile into .LNK files that the linker processes into .LOD files. These .LOD files can be converted by a utility into .EXE files; otherwise, the user can call the runtime system M2.EXE to load .LOD files and execute them. The compiler generates listing files by the user setting a switch on the command line. In addition, .REF files are produced for use by the debugger.

The code model appears to be an implementation of the full Intel/Microsoft standard. Procedure activation records are implemented through the use of ENTER and LEAVE instructions when compiling to an 80186/286. The code is very fast, as evidenced in the results for the benchmark tests. In looking at the code using SYMDEB, however, the compiler appears to reload its segment registers more often than is necessary simply because the compiler has no mechanism to detect that correct values are already there.

Both versions of this compiler support the definition of Modula-2 as outlined in the second edition of Wirth's book; the third edition was published after the compilers were released. (Logitech is reportedly preparing an update to meet this most current specification.) Even so, the only significant difference between the compiler and the current definition of the language is the compiler's requirement that definition modules explicitly export their identifiers. The system's low-level facilities are complete, with absolute addresses for



A Contradiction!

Running Under PC DOS

>750 million bytes formatted in two volumes for the "Eagle" (one volume/drive)...M2361A can hold 552MB/volume...data transfer rate up to 2.4MB/sec....data access time - 18ms/drive...variable interleave capability...partitioning possible...drives built to mainframe specifications with mainframe reliability...greater than 20,000 hours MTBF.

For further information contact:

Upper Bound Micro

18 Elizabeth Street, W. Conshohocken, PA 19428
(215) 825-0505 FAX (215) 828-8618

The "Eagle" is a trademark of Fujitsu America, Inc.
PC DOS is a trademark of IBM.

CIRCLE NO. 169 ON READER SERVICE CARD

variable, interrupt access, in-line code generation, and CPU register access.

Logitech's libraries are quite extensive and provide binary-coded-decimal (BCD) support, interrupt handling, DOS services, random file I/O, a mouse interface, multitasking, and serial device handling. Notably absent, however, is screen graphics support.

This package offers the user three options for debugging. First, in the basic system, an execution error simply produces a RAM dump. A postmortem debugger interprets the RAM dump and pinpoints the error. Second, the library module Debug, if imported, traces the stack. Finally, a powerful runtime debugger is also available.

The Logitech documentation is well done. The manual is now perfect-bound and includes a brief tutorial on the language. The index is complete, and the definitions of the libraries are reprinted in the manual for easy reference. In short, this is a very good compiler and a leader in this field, but it is in need of some minor burnishing.

Modula Corporation. For a long time the source of a p-code (pseudocode) implementation of Modula-2, this company has recently begun to produce a native code implementation for the PC, called PC Modula-2. The system includes a compiler, a loader, and a linker; a windowing, source-level debugger is available at extra cost. However, an editor is not offered with this product.

The compiler is invoked from DOS; its output is a .RLX file containing the relocatable object code, and a .RFC file with information for the debugger. The compiler error messages are terse, but adequate. The user can control runtime checking through the use of switches on the command line.

In its code model, the Modula Corporation compiler limits procedure parameter lists to 16 words, which can cause some problems.

The language that is implemented by this compiler conforms to the third edition of Wirth's report: definition modules do not have to export their identifiers explicitly, priority modules and multitasking are available, and strings are not a predefined type of the language but are arrays of characters terminated by a null character.

This system's low-level facilities include CODE procedures (which are short in-line assembly language routines), implementation modules written in assembly language, absolute addressing, and I/O device management. Note, however, that the assembly language-level implementation modules must follow a rigid format in order for the linker to load them properly.

The code model for the Modula Corporation compiler is not the Intel/Microsoft standard. The compiler limits procedure parameter lists to a total of 16 words in length, which can be a problem—open arrays require 3 words of parameter space for the address and the size of the array, VAR parameters require 2 words for the address, and value parameters require as many words as their base types are large.

This Modula-2 compiler does not use the RET instruction to return from a subroutine. Instead, the compiler emits an indirect jump instruction, taking its destination address from the stack at the point where the CALL instruction had placed the return address, and leaving the calling routine to clean up the stack. The performance overhead exacted by this process is evidenced in the procedure call benchmarks.

The results for these tests can be puzzling without the knowledge that this compiler was ported from another machine and, more significantly, that the chosen code model underemploys the instruction repertoire of the 8086. This is evidenced by the fact that the Modula compiler stores its pointers with the segment portion in the low-order word—in reverse of the Intel standard. This prevents the compiler from using the pointer loading instructions, LDS and LES.

The generated code is also inefficient at times. In setting up a CASE statement, for example, the address of the selected code is loaded from the jump table into a register and a jump is taken relative to the register rather than out of the table itself.

However, the balance of its scores in the benchmarks are very good. This compiler turned in the shortest time for the cardinal arithmetic test. (During the real number test, though, the library halted execution of the benchmark with an error message claiming an attempted conversion of too large a floating-point number into an integer. It is unclear why **sqrt(sin(ln(pi)))** would require an integer conversion at all, even for an intermediate value.)



CIRCLE NO. 101 ON READER SERVICE CARD

The libraries for this compiler include Wirth's standards and are otherwise comprehensive. In addition, the system supports multitasking, string management, graphics, and long sets (those with 65,534 elements). The debugger is an interesting one, with multiple screens that permit the user simultaneous examination of the stack, global variables, and registers. The Modula Corporation documentation is very thorough and well indexed.

PCollier Systems. This consulting firm has produced a very compact implementation of the language. The Modula-2PC package includes compiler, editor, and linking loader. The editor is a standard full-screen utility with its own set of keystroke commands that are unlike any other and do not use the PC's function keys. It communicates with the compiler through the DOS environment; error information is passed back to the editor for cursor positioning.

The language implementation is complete as specified in the third edition of the language report. The WHILE statement is extended to permit multiple Boolean expressions and statement sequences. For example, in

```
WHILE
  (negative) DO j := j + 17 |
  (i > 0) DO j := j – 17
END
```

the control sequence loops through conditional expressions until the first true expression is encountered. Its associated statements are executed, then control returns to the top of the WHILE loop. The loop terminates when all of the conditional expressions are false.

PCollier's low-level facilities are adequate. Its CODE procedures are macros for small sequences of machine instructions. Once defined, these CODE procedures can be invoked simply by naming them; they will be expanded into in-line code at that point in the program. The CODEGEN pseudo-procedure, however, is closer in concept to Turbo Pascal's INLINE statement, and permits the generation of lengthy assembly language sequences. External .COM files can be included in the object code using the LOADASM procedure. Register access in Modula-2 statements is not allowed. Software interrupts can be called through a procedure imported from the DOS module.

The real number support provided by the system depends upon which of two included versions of the loader is used to create the program: one uses the 8087 coprocessor, the other does not. The 8087 "detect or emulate" decision cannot be postponed until runtime. The loader will dynamically link to imported modules as they are called; it also can generate a .EXE file.

The code model for this compiler differs from the Intel/Microsoft standard. First, the process of calling procedures is handled by a global module manager. The manager is passed a key for the requested module. The targeted procedure is located and control passed to it. If the targeted procedure is not present in RAM, it is located on disk and loaded. Second, arguments are passed on the stack to the module manager, which then removes the parameters from the stack and places them into the data segment for the routine. If a recursive call is made, the previous invocation's data segment is preserved and the current set of parameters is instantiated. This is not very efficient, as the benchmarks show.

The code generated by this compiler for handling Boolean expressions inside IF statements is different from most compilers. This generated code calculates a Boolean value for the expression and tests it against TRUE, jumping on the result of the comparison. The calculation requires one set of jumps; the test is an additional set. This is a less efficient method than that of the other compilers, which emit direct comparisons and jumps.

PCollier's arithmetic code is also less efficient than that of the others. Modula-2PC uses the 8086's BX register for interfacing to RAM. Note (in figure 2) the two MOV instructions at the end of the code: one moves the result into BX, the other moves BX into RAM.

The libraries for this compiler are complete and include a nice screen, window, and menu management module. The Terminal module, however, does not follow the standard: Terminal.Read takes its first input from the DOS command line. Coroutines are supported, but multitasking is not.

Clearly, the strength of the PCollier implementation is its documentation. The book is exceptionally well written and indexed. With its fine tutorial, it easily could retail as an introduction to Modula-2 on its own.

Pecan Software Systems, Inc. This company acquired Volition Systems and subsequently upgraded and rereleased Volition's Modula-2 product under its own label. Pecan Modula-2 operates in the Power System environment, Pecan's new name for the UCSD (and later SofTech) p-System, a virtual machine environment that has existed since the mid-1970s. This compiler is invoked from

PROTECT YOUR COPIES OF PC TECH JOURNAL

Make your collection of PC TECH JOURNAL a handsome addition to your office or home—and protect and organize them for easy reference!

PC TECH JOURNAL Magazine Binders and cases are made of durable luxury-look leatherette over quality binder board. Custom designed for PC TECH JOURNAL, every order receives FREE transfer foil to mark dates and volume numbers.

FOR FAST SERVICE CALL TOLL-FREE 1-800-972-5858

MAGAZINE BINDERS

Hold your issues on individual snap-on rods. $8.95 each; 3 for $25.75; 6 for $48.75.



OPEN BACK CASES

Store your copies for individual reference. $7.95 each; 3 for $22.95; 6 for $43.95.



PC TECH JOURNAL
P.O. Box 5120
Philadelphia, PA 19141

Please send ☐ Binders ☐ Cases Quantity _____

Payment enclosed $_____.* Add $1 per order for postage & handling. (Outside USA, add $2.50 per unit ordered, US currency only.)

Charge my:
☐ Amex ☐ Visa ☐ MC (Minimum order $10.)

Card No._____ Exp. Date_____

Mr./Mrs./Ms._____
please print full name

Address_____

City_____

State_____ Zip_____

*PA residents add 6% sales tax.

the Power System and emits either p-code or native 8086 code. The p-code is threaded code for execution by the Power System's virtual machine interpreter. Pecan has ported the Power System and its virtual machine to several different computing environments.

The Power System runs above DOS, but does not use DOS for its file operations. File I/O is performed directly by the Power System, using an antiquated sector system that requires allocating a given number of sectors to a file at the time of a file's creation. Deleting a file on a Power System virtual volume leaves a "hole" of unused sectors the same size as the deleted file. When many create and delete cycles occur on a volume, it can become extremely fragmented, and free sectors need to be collected into a single block by an operation called Krunch.

The editor is a fairly standard text entry model. Provisions are included for editing multiple files, with text transfer between files, function key macros, and automatic indentation. This editor can perform source code logging, which automatically inserts a comment into the source file, containing the date and a line of explanation for changes to the program source code.

Unlike UCSD Pascal, Pecan's Modula-2 editor does not have an interface to the compiler; therefore, the user must save the work in progress and exit the editor before invoking the compiler. Compile-time errors are not communicated to the editor for cursor positioning in the source file.

The compiler accepts standard Modula-2 syntax according to the third edition of Wirth's report, including module priority. Errors are correctly spotted and the associated messages are informative. Compiler options are embedded as comments within the text of the module being compiled.

Pecan has added several extensions to the Modula-2 language in this implementation, mostly to accommodate features of the Power System. BCD variables are supported and are defined using the predefined type PACKED. Function procedures may return any type as a result, including RECORDs.

However, some restrictions cloud the differences between CARDINAL and INTEGER variables. For example, cardinal division cannot be performed when an operand is greater than 32,767. Similarly, case labels that are of type CARDINAL cannot exceed 32,767.

This system's low-level facilities are geared toward p-code: no direct access to the host system is allowed. The user can generate p-code routines using several pseudo-procedures.

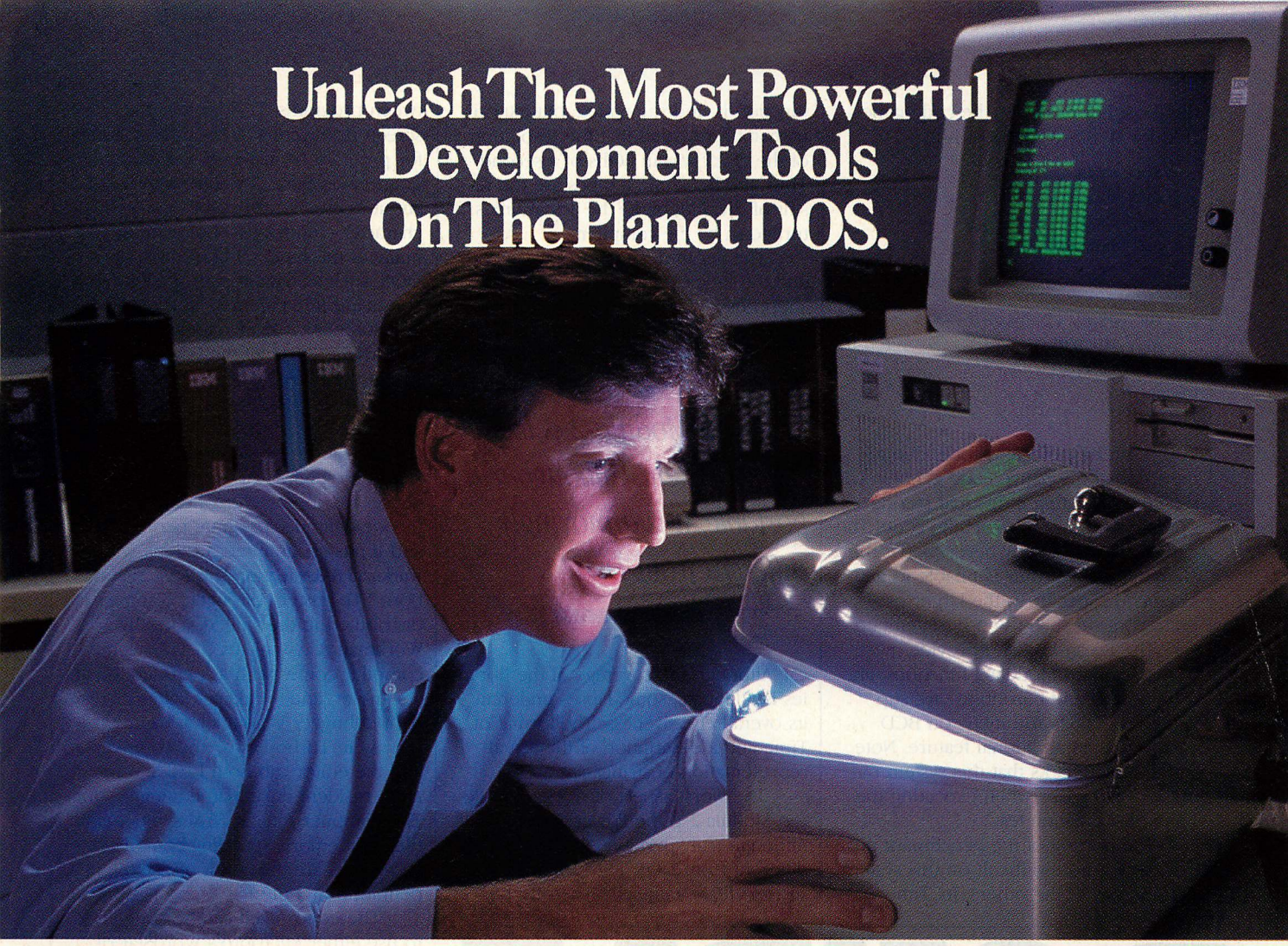
In its default mode, this compiler directly emits p-code into Power System .CODE files for implementation and program modules. Definition modules compile to .SYM files, which provide the necessary symbolic information to the compiler when the module is imported. The native code generator is invoked by setting a compiler directive option. Configuring the Power System for an 8087 requires changing the name of the default code generator and Power System runtime library files. Once so configured, the 8087 is required to be present for a floating-point program to run; otherwise, the program will hang the computer.

The performance of the native code generator can be seen in the benchmark results and in figure 2. This compiler, like the PCollier product, actually calculates a Boolean value for simple comparison expressions, then tests the state of the Boolean value, rather than simply comparing the two operands and branching on the state of



CIRCLE NO. 185 ON READER SERVICE CARD

Unleash The Most Powerful Development Tools On The Planet DOS.



UNIFY DBMS/DOS. The UNIX World Leader Brings A New Dimension To DOS Application Development.

What happens as the DOS world expands? As a new generation of hardware takes over? As networking becomes more important? The potential is enormous. But until now, the tools to achieve it have been limited.

Now a leader from another world unleashes that potential: UNIFY® DBMS. The leading relational DBMS in the UNIX™ world. And now, the most advanced set of application development tools in the DOS world.

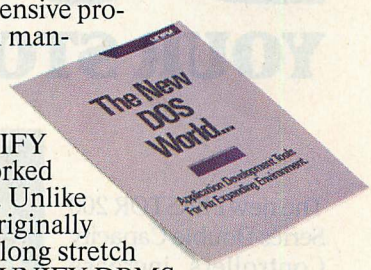
With UNIFY DBMS, DOS developers have new power to build more sophisticated applications than ever before possible.

The power to write high performance "C" programs that will access the data base, using Unify's Direct Host Language Interface.

The power of an industry standard query language—SQL.

The power of unmatched speed in production applications. Only UNIFY DBMS is specifically engineered for transaction throughput. With unique performance features like PathFinder™ Architecture multiple access methods, for the fastest possible data base access.

The power of comprehensive program development and screen management tools. Plus a state-of-the-art fourth generation report-writer.



What's more, with UNIFY DBMS, the potential of networked applications becomes a reality. Unlike DBMS systems which were originally single-user (and which have a long stretch to accommodate more users), UNIFY DBMS is a *proven* multi-user system.

And because UNIFY DBMS/DOS is the best of two worlds, it offers you the most powerful benefit of all: DBMS applications that can grow as your needs grow. From single user DOS. To networked DOS. To multi-user UNIX. All without changing your applications.



Call the Unify Information Hotline for our free booklet: The New DOS World.
(503) 635-7777

See Us At
UniForum
January 20 – 23, 1987
Washington Convention Center
Washington D.C.

UNIFY CORPORATION
4000 Kruse Way Place
Lake Oswego, OR 97034

© 1986 Unify Corporation. UNIX™ is a trademark of AT&T Bell Laboratories.
CIRCLE NO. 156 ON READER SERVICE CARD

the flags. For example, the code generated for IF i > 0 THEN ... is

```
    xor  ax,ax
    mov  dx,i
    cmp  dx,0
    jnz  notzero
    inc  ax
notzero:
    shr  ax,1
    jc   truepart
    jmp  falsepart
truepart:
    ...
```

The result in inefficient code, due to the fact that the native code generator is directly translating individual p-code operations into 8086 machine code. The native code generator never "sees" the source code and thus is unable to generate code specific to the larger program context, such as an IF statement.

The Pecan libraries provide most of the standard Modula-2 functions, and multitasking and coroutines are supported. Decimal arithmetic on BCD variables is an additional feature. Note in the benchmarks that this system ran out of heap space when executing the linked list creation test.

The porting of the benchmark programs was more difficult on this system than on any of the others. Its Time function is undocumented; but experimentation revealed that the resolution of the clock was in 60ths of a second. Thus, the benchmark times are the result of dividing by 60 the actual times turned in by the compiler.

The Pecan documentation is good, and the index is thorough, but it presupposes a working knowledge of Pascal. Although the coverage of the libraries is adequate, some lapses undercut its overall effectiveness. The format of Time and a description of the actual representation of strings are absent.

Clearly, a restrictive aspect of Pecan Modula-2 is the Power System. Although truly portable, it is annoyingly different from DOS.

The worst aspect of working in Pecan Modula-2 is the Power System itself. Although truly portable, it is annoyingly different from DOS. For example, the screen clears, rather than scrolls, between programs. This inhibits the screen from acting as short-term storage for program results across several program invocations. The user interface is about as clumsy as that of DOS—one-character commands—but without many of the redeeming features of DOS, such as default path specifications, command line parameters to programs, and automatic file maintenance.

The strength of the Power System lies in the diversity of machines to which it has been ported. With Pecan Modula-2, true portability can be achieved if the programmer limits himself to the high-level features of the language. For low-level applications, the p-code degrades program performance and limits programmer options.

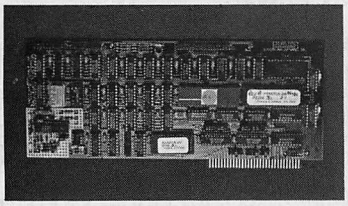
Workman & Associates. The Australian-born FTL (faster than light) compiler is distributed in the United States by Workman & Associates. This powerful compiler comes with a useful editor, an assembler, and a linker. A memory-resident debugger is an added feature.

As in Turbo Pascal, the compiler and the editor can both load and remain resident during an editing session. The compiler retains .SYM symbol table files in memory, speeding compilation. (.SYM files are the result of compiling a definition module.)

The editor itself can be configured to use either Micropro's WordStar or Unipress Software's Emacs control key sequences for commands. The user can set up keyboard macros, but language constructs cannot be inserted by Alt-key combinations, as they can be in the ITC editor. The editor is limited to three fixed-size windows containing up to three simultaneously opened files. The user can set compiler options from a screen in the editor, and the compiler can be called up quickly. The linker is easily accessed from the editor as well, but promptly runs out memory in all but the largest machines.

The compiler is (as claimed) very fast, with a compile time benchmark of 13 seconds. This one-pass system generates code for a small memory model: 64KB of data and 64KB of code. The compiler output is in the form of .SMR files, which are fed to the linker. The code it generates is reasonably tight, and the times for the FTL compiler in the benchmarks reflect the advantages of the small memory model. The code model is similar to Intel/Microsoft: Parameters are pushed onto the stack from left to right. The called procedure cleans up the procedure activation record. Functions return their results on the stack rather than in registers. CASE

DOUBLE
YOUR STORAGE CAPACITY



The new PERSTOR 200 Series Double Capacity Controllers increase the storage capacity of your ST506/412 Winchester hard disks by 90% or more. Advanced RLL encoding technology is used to increase data transfer rate to 9 and 10 megabits per second, and a 56 bit error correction code is used to assure data integrity. What's more, it works with both RLL and MFM drives with plated or oxide media.*

To place an order or become a dealer call (602) 948-7313.

PERSTOR

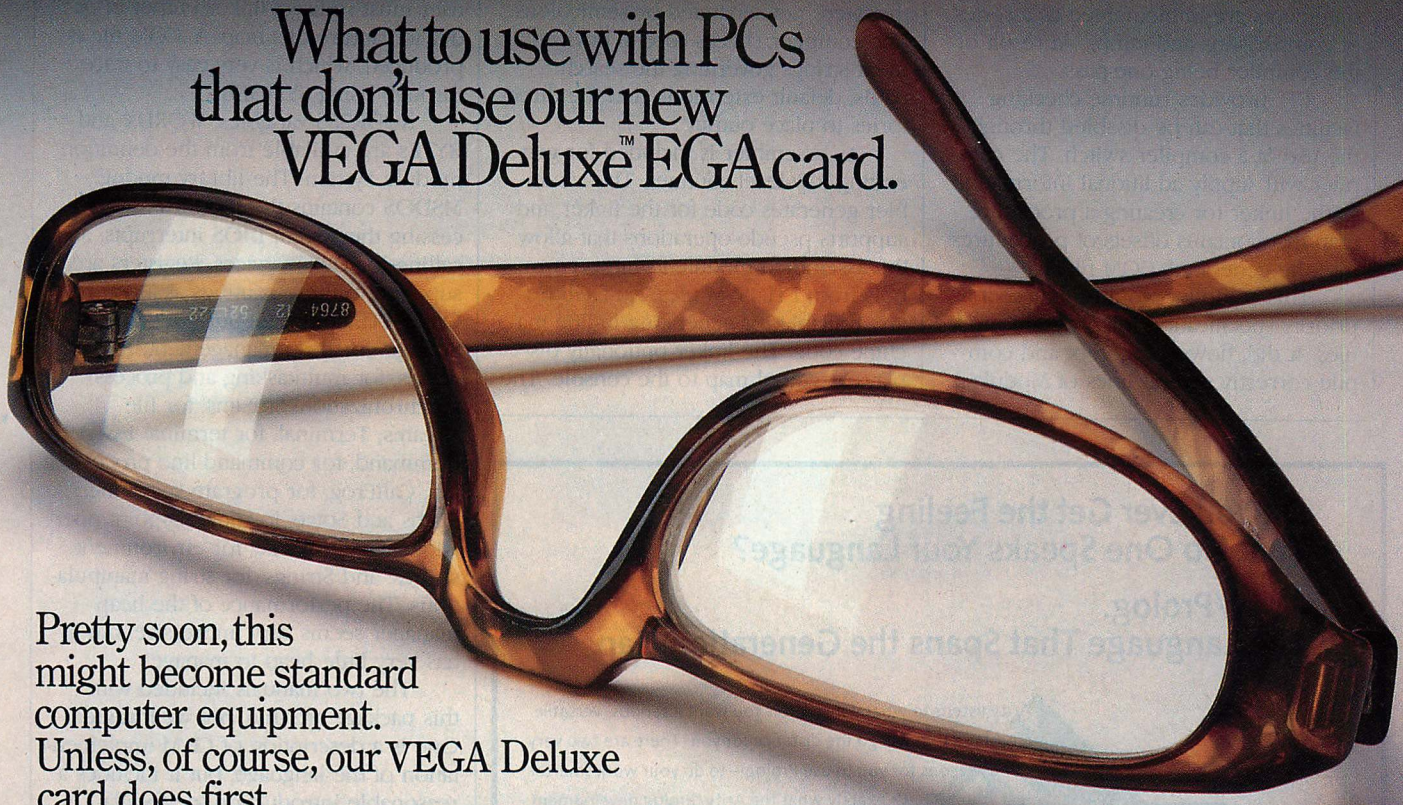
Sensible solutions for your hard disk problems.

Systems and Software, Inc.
7825 East Redfield Road
Scottsdale, Arizona 85260

*call for specific drives.

CIRCLE NO. 231 ON READER SERVICE CARD

What to use with PCs that don't use our new VEGA Deluxe™ EGA card.



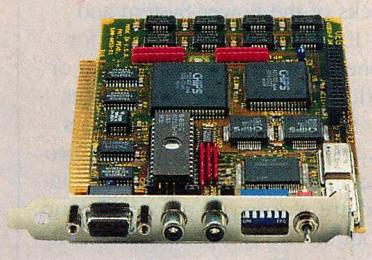
Pretty soon, this might become standard computer equipment. Unless, of course, our VEGA Deluxe card does first.

And that's a distinct possibility. After all, this short-card video adapter offers 37% higher screen resolution than standard EGA cards.

Visualize it: your existing software sharper and clearer by more than a third. Programs like Lotus 1-2-3 and Symphony, with 132 columns and 43 lines. Microsoft Windows in 640 x 480, too. And AutoCAD. As well as EASYCAD, EGA Paint, GEM, Dr. Halo II, In·A·Vision, Windows Draw, Windows Graph, and much more.

In fact, with all this information on the screen, there's only one thing there's less of. Strain on your eyes.

That's because the VEGA Deluxe gives you both 640 x 480 *and* 752 x 410 resolution. (Of course, for more than 640 x 350, the VEGA Deluxe requires a Multisync® or equivalent.)



The VEGA Deluxe is also compatible with every other video standard: EGA, CGA, Hercules and MDA. And when used with a compatible monitor, it automatically selects which mode is right for your software's needs.

At Video Seven, we've sold over 200,000 EGA cards to people who've seen the difference. If you'd like to see the difference for yourself, just call 1-800-238-0101 for the name of the Video Seven dealer nearest you. (In California, call 1-800-962-5700.)

The VEGA Deluxe. It puts everything in proper focus.

Video Seven Inc., 550 Sycamore Drive, Milpitas, CA 95035.

VIDEO SEVEN

We make a clear difference.

High resolution modes require TTL color monitors capable of 25 KHZ and 29.4 KHZ. Trademarks: VEGA Deluxe—Video Seven Inc., Hercules—Hercules Computer Technology, MultiSync—NEC Home Electronics (USA) Inc., EASYCAD—Evolution Computing, EGA Paint—Rix SoftWorks Inc., GEM—Digital Research Corp., Dr. Halo—Media Cybernetics, In·A·Vision, Windows Draw, Windows Graph—Micrografx Inc.; Registered trademarks: Video Seven—Video Seven Inc., Lotus 1-2-3, Symphony—Lotus Development Corp., Microsoft—Microsoft Corp. Video Seven reserves the right to change specifications without notice.

CIRCLE NO. 204 ON READER SERVICE CARD

statements are implemented as a series of comparisons and jumps—a result of the compiler being one-pass.

FTL provides runtime checking facilities that can be disabled through the use of a compiler switch. The compiler will supply additional information to the linker for creating a procedure map that contains offsets of procedures into the generated .COM file.

This compiler would not accept a priority specification for program modules. It did, however, accept and compile correctly several lines of Modula-2 that were not separated by semicolons. The compiler checks several environment strings governing the search paths, default extensions, and the directories to place output.

An assembler is included for generating low-level routines. The assembler generates code for the linker and supports pseudo-operations that allow the assembled module to import symbols from FTL .SYM files. No facilities are available for generating in-line machine code. The linker optionally displays a symbol map to the console. The user must redirect linker output in order to capture this map. A .COM file is produced, which is very easy to trace through using SYMDEB.

ADDRESS variables, WORDs, and BYTEs are available from the definition module System. The library module MSDOS contains the routines for accessing the system BIOS interrupts, and calling for DOS services. Registers are supplied to these routines as a record. Other FTL library modules include Storage, for heap management; Processes, for multitasking and process synchronization; Streams, for file streams; Terminal, for terminal I/O; Command, for command-line processing; CallProg, for program spawning; Maths and Solve, for floating-point operations; GetEnvName, for environment strings; and Strings, for string manipulations. The performance of the heap manager seems very impressive, but it has very little heap to manage.

The two manuals included with this package are thin, but well written. One is a description of CP/M implementation of the language, but it includes a reasonable introduction to Modula-2. The other describes the DOS version of the system, and makes frequent references to the CP/M manual.

The source code for the editor is available at extra cost. Despite the limitations of the small model, this compiler is a good value and offers an adequate introduction to the language.

MODULAR DIRECTIONS

The six compilers reviewed here present the project manager with a variety of choices. If portability across a number of machines is important, the obvious choice is Pecan, in spite of the drawbacks of the Power System and p-code. If a manager is designing for PCs alone, then the Logitech system or the Modula Corporation compiler fits the bill. For first-time Modula-2 users, the FTL and the PCollier compilers provide excellent introductory documentation and are reasonably priced. The jury is still out on the ITC product, but it is bound to improve.

Their overall quality notwithstanding, none of these compilers puts Modula-2 in a position to challenge the supremacy of C in PC programming (yet). Only when these products allow foreign object code integration, using the Microsoft standard, will Modula-2 begin to take its true place in the PC market. In another direction, a Modula-2 compiler that incorporates Microsoft Windows capability would, of course, prove a noteworthy step. The development of

Do You Ever Get the Feeling That No One Speaks Your Language?

Arity/Prolog. The Language That Spans the Generation Gap.



Arity listens to what you ask for. You want a serious, versatile language that will go the distance for you. There are two very good reasons for you to use Prolog—to do your work smarter and faster. That's exactly what the Arity/Prolog development environment will help you do. Our powerful tools, based on the general purpose programming language Prolog, will significantly reduce your development time and allow you to solve a wide range of application problems.

No translation required Our development environment for the IBM PC family and all MS-DOS compatibles includes the Arity/Prolog Compiler and Interpreter, the Arity/Expert System, and Arity/SQL. And you can tie them all together. You can interface with several other programming languages and build extensions to your existing applications. You'll be truly multilingual—what better way to span the generation gap?

It can take you to new places You'll discover amazing speed, power, and flexibility using the Arity/Prolog programming environment, with its one gigabyte of virtual memory and fast, compact compiled code, for conventional applications. And if you're working in new territories, like expert systems or sophisticated database management systems, you'll be speaking the native tongue.

Speak it freely Our products are not copy protected and we charge no royalties, so you can use them in as many end-user applications as you'd like. Why keep the language of solutions all to yourself?

Join the thousands of assembly and C programmers who already use Arity/Prolog—the language of solutions.

Call 1-800-PC-ARITY Today.

Masschusetts residents call 617-371-1243.

Software that roars.

ARITY CORPORATION
30 Domino Drive, Concord, MA 01742 U.S.A.
1-800-722-7489 or in Massachusetts call 617-371-1243

CIRCLE NO. 136 ON READER SERVICE CARD

Turn Your System/3X Into The Perfect Host With PCOX Technology.

PCOX™ 5250 products make your System/3X treat your PCs like members of the family.

Your System/3X and your PCs already live together. Now they can work together, too. Thanks to PCOX Technology.

With PCOX 5250 connections, your PCs enjoy the full privileges of a 5251 Model 11, 5291 or 5292.

Which means your PCs can access and transfer files from your System/3X data base, use its host as a departmental processor, or participate in your company's distributed SNA network.



MORE WAYS TO SAY HELLO. The PCOX 5250 series comes in twinax and remote versions—one for local connections, and one for connections over phone lines.

PCOX 5250 products come in twinax and remote versions, so PCs can enjoy S/3X connections in person or over phone lines.

And both versions support up to seven concurrent host sessions. No competing product delivers more.

So let PCOX Technology open doors between your System/3X and your PCs.

Call CXI today, toll-free.

800-225-PCOX.

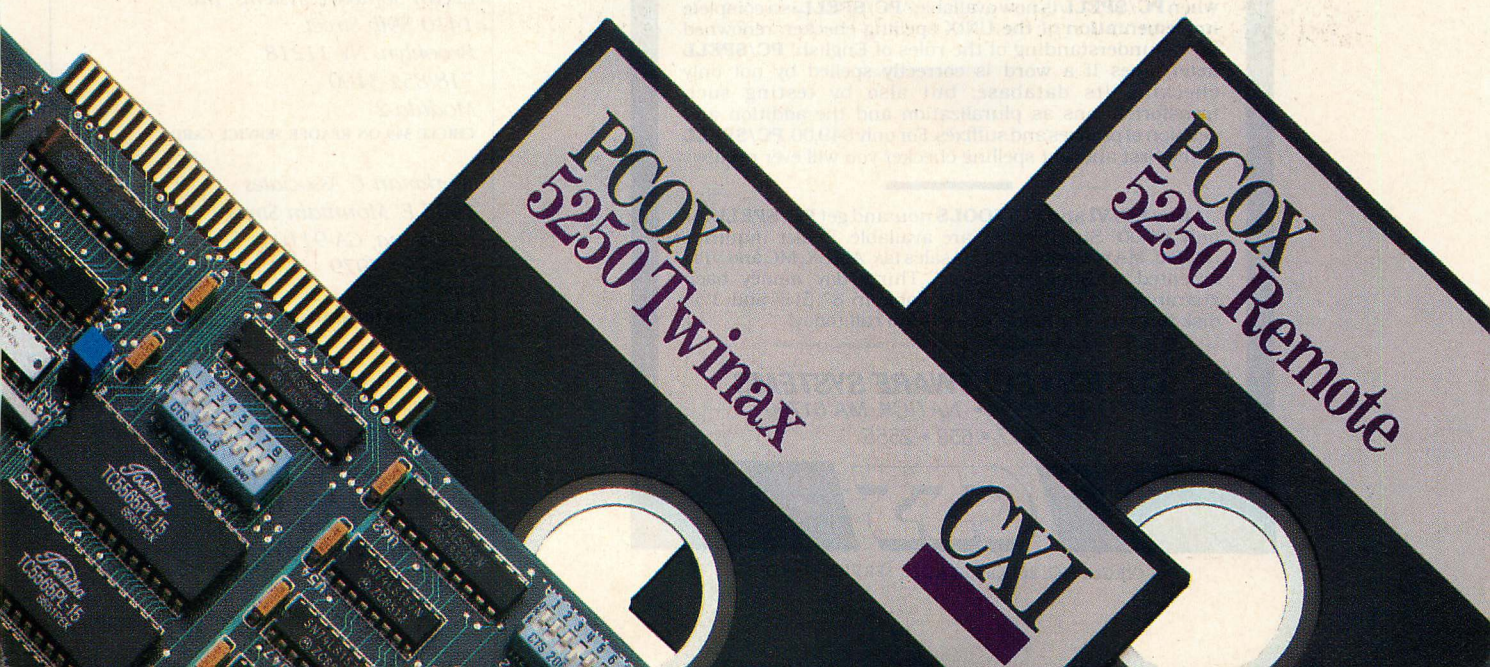
In California, call 415-424-0700.

CXI

CXI, Inc., 3606 West Bayshore Road
Palo Alto, CA 94303. Telex: 821945

PCOX and all PCOX products are trademarks of CXI, Inc.

CIRCLE NO. 217 ON READER SERVICE CARD



PC/VI™

UNIX's VI Editor Now Available For Your PC!

Are you being as productive as you can be with your computer? An editor should be a tool, not an obstacle to getting the job done. Increase your productivity today by choosing **PC/VI** — a COMPLETE implementation of UNIX* VI version 3.9 (as provided with System V Release 2).

PC/VI is an implementation of the most powerful and most widely used full-screen editor available under the UNIX operating system. The following is only a hint of the power behind **PC/VI:**

- Global search or search and replace using regular expressions
- Full undo capability
- Deletions, changes and cursor positioning on character, word, line, sentence, paragraph, section or global basis
- Editing of files larger than available memory
- Shell escapes to DOS
- Copying and moving text
- Macros and Word abbreviations
- Auto-indent and Showmatch
- MUCH, MUCH MORE!

Don't take it from us. Here's what some of our customers say: "Just what I was looking for!", "It's great!", "Just like the real VI!". "The documentation is so good I have already learned things about VI that I never knew before." — *IEEE Software*, September 1986.

PC/VI is available for IBM-PC's and generic MS-DOS† systems for only $149. Included are CTAGS and SPLIT utilities, TERMCAP function library, and an IBM-PC specific version which enhances performance by as much as TEN FOLD!

PC/TOOLS™

What makes UNIX so powerful? Sleek, Fast, and **POWERFUL** utilities! UNIX gives the user not dozens, but hundreds of tools. These tools were designed and have been continually enhanced over the last fifteen years! Now the most powerful and popular of these are available for your PC! Each is a complete implementation of the UNIX program. Open up our toolbox and find:

• BFS	• DIFFH	• OD	• STRINGS
• CAL	• DIFF3	• PR	• TAIL
• CUT	• GREP	• SED	• WC
• DIFF	• HEAD	• SEE	

All of these for only $49.00; naturally, extensive documentation is included!

PC/SPELL™

Why settle for a spelling checker which can only compare words against its limited dictionary database when **PC/SPELL** is now available? **PC/SPELL** is a complete implementation of the UNIX spelling checker, renowned for its understanding of the rules of English! **PC/SPELL** determines if a word is correctly spelled by not only checking its database, but also by testing such transformations as pluralization and the addition and deletion of prefixes and suffixes. For only $49.00, **PC/SPELL** is the first and last spelling checker you will ever need!

Buy **PC/VI** and **PC/TOOLS** *now* and get **PC/SPELL** for only $1.00! Site licenses are available. Dealer inquiries invited. MA residents add 5% sales tax. AMEX, MC and Visa accepted without surcharge. Thirty day money back guarantee if not satisfied! Available in 8", 5¼" and 3½" disk formats. For more information call today!

*UNIX is a trademark of AT&T †MS DOS is a trademark of Microsoft.

CUSTOM SOFTWARE SYSTEMS
P.O. BOX 678 • NATICK, MA 01760
617 • 653 • 2555

CSS

UNIX TOOLS FOR YOUR PC

UNIX TOOLS FOR YOUR PC

CIRCLE NO. 261 ON READER SERVICE CARD

MODULA-2

a Windows application, with its large library of routines and lengthy header file, would be speeded up considerably by taking advantage of Modula-2's separately compiled modules. The strength of this language is in its ability to spot type mismatches in procedure calls—this is a situation that often generates runtime errors in C.

Modula-2 represents the future in large-scale programming—the development of a standard is important. The software engineering features that it contains are shared only by Ada. Given the expense of creating software, it seems inevitable that Modula-2 will become the language of choice for many software developers.

Interface Technologies Corporation
3336 Richmond, Suite 200
Houston, TX 77098
800/922-9049; 713/523-8422
M2SDS and SDS-XP
CIRCLE 345 ON READER SERVICE CARD

Logitech, Inc.
805 Veterans Blvd.
Redwood City, CA 94063
415/365-9852
Modula-2
CIRCLE 346 ON READER SERVICE CARD

Modula Corporation
1673 W. 820 North
Provo, UT 84601
801/375-7400
PC Modula-2
CIRCLE 347 ON READER SERVICE CARD

PCollier Systems
7925-A N. Oracle Road, Suite 390
Tucson, AZ 85704
800/522-2060; 800/654-7396
Modula-2PC
CIRCLE 348 ON READER SERVICE CARD

Pecan Software Systems, Inc.
1410 39th Street
Brooklyn, NY 11218
718/851-3100
Modula-2
CIRCLE 349 ON READER SERVICE CARD

Workman & Associates
1925 E. Mountain Street
Pasadena, CA 91104
818/791-7979
FTL
CIRCLE 350 ON READER SERVICE CARD

John T. Cockerham, M.D., is a cardiologist at The Children's Hospital in Boston and is on the faculty of Harvard Medical School. His most recent articles for PC Tech Journal *were "Evaluating the EGA: The EGA Standard," October 1986, p. 48, and "Evaluating the EGA: The EGA Spectrum," October 1986, p. 80 and November 1986, p. 147.*

Here's a plug for dBASE.™

Clipper™ NEWS RELEASE

FOR IMMEDIATE RELEASE

For information contact:
Nantucket
Nantucket Corporation®
12555 Jefferson Blvd.
Los Angeles, CA 90066
(213) 390-7923

CLIPPER™ NETWORKS DBASE APPLICATIONS

LOS ANGELES, California...Nantucket's Clipper now lets developers and business persons plug an unlimited number of workstations together to run their dBASE III and dBASE III PLUS applications, using Clipper's new networking capabilities.

This new release compiles programs to run on networks that support DOS 3.1 calls for networking functions, plus single-user programs for DOS 2.0 or greater.

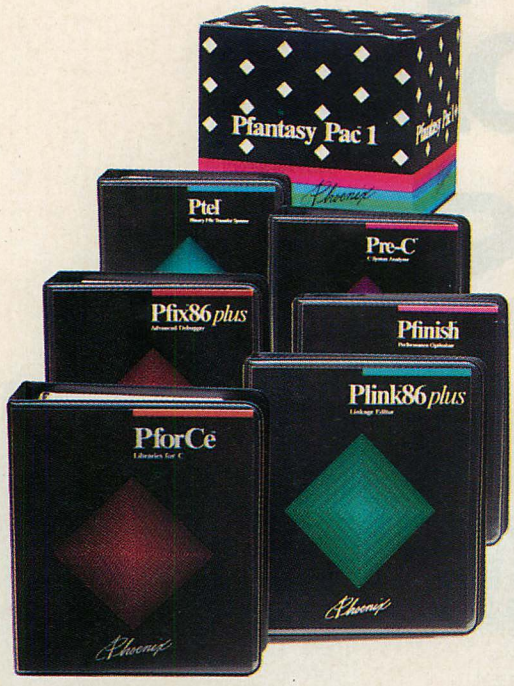
Compiled Applications can be distributed freely, need no runtime module, no licensing fee or royalty. And there is no extra cost per user, regardless of how many users are connected to a Clipper network. Plus the new release now packs even more of Clipper's famous speed, on both single-user and networking applications.

The new Clipper also sports Expanded Memory support, additional functions and improved memo fields. The new release, dubbed Autumn '86, is not copy protected.

Clipper Autumn '86 is available for a suggested retail price of $695. Registered users of Clipper may upgrade to the new version for $139.

Clipper and Nantucket are trademarks of Nantucket Corporation.
Other products trademarked by others. Copyright 1986, Nantucket Corporation.

CIRCLE NO. 224 ON READER SERVICE CARD



PC BRAND NOW PFEATURES PFANTASTIC

Phoenix

Today's professional programmer demands high performance tools that speed up and enhance the application development process. That's why more programmers rely on one company for the best engineered, highest performance tools available—Pheonix.

Phoenix offers a full line of powerful, yet easy-to-use, tools that help programmers more efficiently write, test and deliver the best applications possible. Complete, fully detailed documentation accompanying each tool enables quick mastery of the product. In addition, all Phoenix tools are backed by full-time customer support professionals respected throughout the industry for their commitment and expertise. As a result, many Phoenix tools are already established as industry standards and others are fast on their way.

Plink™ 86*plus*

Plink86*plus* is the only linkage editor containing advanced overlay capabilities. It handles any compiler or assembler producing standard Intel or Microsoft OBJ files, including COBOL and FORTRAN, Lattice C, CI C-86, Microsoft/IBM languages, and mbp/COBOL. Virtual memory management ensures ample capacity for symbol and common block names (35,000). Plink86*plus* supports an unlimited size file, an unlimited number of modules and up to 4,095 overlays nested up to 32 deep. Merges object modules, caches overlays in extended or expanded memory, and automatically reloads overlays upon function return. Includes Plib86 object library manager.
List: **$495** PC Brand: **$325**

Plib™ 86

Plib86 is an object module librarian for Microsoft and Intel format object modules and libraries. With Plib86, you can add, delete or extract modules to or from libraries, and explode a library to its component modules with one command. Plib86 also produces program cross-reference listings in various formats to a disk file, screen or printer.

Originally listed at $195. Plib86 is now included with Plink86*plus*.

Ptel

A Pfantastic communications program for use with most popular modems such as Hayes and compatibles, DEC, Racal Vadic, Anchor, US Robotics and Novation. Ptel automatically adapts to Telink, XModem, Kermit or Modem 7 for CRC checking and for ufn and afn (i.e. "wildcard") file name list transfers, if the bulletin board or the other end computer supports them. With Telink, you'll even get a forecast of file transmission time, always useful in controlling telephone expenses.

In addition to saving received files, Ptel can create and save a transcript of the session commands and messages.

Highly configurable, with choices temporary or saved as altered defaults. Unique telephone directory of bulletin boards and other services, allowing access by service name. Ptel runs fully interactive or can be batch driven from a script. You can even exit to DOS move files around or run another application and then return to Ptel, all without dropping the line.
List: **$195** PC Brand: **$115**

Pasm™ 86

Pasm86 provides both quick assembly and superior syntax checking, plus a wealth of other features and utilities to maximize programming productivity. It is a fully MASM-compatible 8086 macro assembler that supports 8087, 80286 and 80287 operating code mnemonics. Features include ability to: define local symbols in the current procedure, assemble files with up to 15,000 symbols, define symbols at assembly time, obtain listings of error lines only, with warning messages on questionable statements. Pasm's comprehensive documentation includes detailed descriptions and examples of each processor instruction. Now includes Pfix-Lite, a subset of Pfix86*plus*.
List: **$195** PC Brand: **$125**

Pmate™

A full screen, single keystroke, fully customizable text processor/editor with advanced features including: ability to run in the background, C and FORTRAN specific macros, automatic disk buffering, ten individual auxiliary buffers, menu, mouse, or command driven with extensive macro command language, and horizontal scrolling. Pmate offers automatic word wrap, text formatting, global or local setting of margins, tab stops, indents, and a unique last-in, first-out "garbage stack" that saves deleted items for recovery.
List: **$195** PC Brand: **$115**

Pre-C™

Now twice as fast! Similar to the Unix LINT for C, but with additional functionality. It crosschecks multiple source files and libraries at once, reporting incorrect, obsolete, and non-portable C usages that no compiler would catch. Pre-C immediately uncovers errors in interfaces between program modules which are very difficult to find using only a debugger. Pre-C accepts full UNIX System III C syntax, a subset of which is implemented by most MS-DOS C compilers, as well as ANSI proposed extensions. External libraries can be used with or without source code. Pre-C libraries for the latest releases of the Mark Williams, Lattice, CI, Microsoft, Wizard, and Aztec "C" compilers are included, and others can be added by supplying Pre-C with function names and arguments. All memory models are supported.
List: **$295** PC Brand: **$155**

Pfinish™

Pfinish helps to "fine-tune" a software product by identifying inefficient or unnecessary sections of code that need to be rewritten for maximum performance. It analyzes your program during execution, producing reports and histograms that give a snapshot of which routines were reached, their callers, how many times each is executed, how much time is spent in each, how many instructions are executed in each, and more. Histograms and tabular reports, sorted by address or symbol, may be written in any page width or height, to a file, the console, or the printer. Pfinish, unlike other "profilers", can use symbol table information to produce much more meaningful analyses on overlays and interrupts.
List: **$395** PC Brand: **$235**

Pdisk™

Pdisk is a complete disk management package that includes advanced Backup/Restore, Tree Management and Disk Cache utilites. Menu, command line or file-driven. Many options permit backup/restore inclusions/exclusions, whole and partial subdirectories, backups by date/time, file type, and backups of all files or files changed since last backup. It can also maintain a log of backups. Supports AT high-density floppies, PC floppies, and any storage device accessible through a device driver. Tree-oriented Directory, Delete, Copy, Compare, and Remove-Directory simplify management of complicated subdirectory structures. CACHE significantly speeds up disk operation on PC/XT/AT by keeping data in memory instead of disk. In addition, CACHE is compatible with the Lotus-Intel-Microsoft (LIM) expanded memory specification, as well as extended memory.
List: **$195** PC Brand: **$125**

Pmaker™

Similar to the Unix MAKE utility, Pmaker keeps track of which modules in a program are changed, and recompiles, reassembles, and relinks those modules to produce a finished product—all with a single command. An essential tool for managing large, complicated, or distributed programming projects, Pmaker is easier to use than similar products, which require you to create lists of all your input files. Pmaker includes a utility that automatically creates and edits such lists based on answers to a few simple questions. Pmaker works with any compiled language, linker, or other tool you use.
List: **$125** PC Brand: **$85**

Pfix™ 86*plus*

Pfix86*plus* is an easy to use, menu driven, multi-windowed symbolic debugger that works with any IBM or Microsoft compiled language. Pfix86*plus* accesses the full symbol table provided by MS Link or Plink86*plus*, and automatically handles Plink86*plus*-overlaid or resident programs. Source code, assembly language translations, stack, data areas, and breakpoints are displayed simultaneously. Features include: In-line assembler for temporary patches, temporary and permanent breakpoint settings, full speed or trace modes, user-assignable variables, dual-monitor support, up to 100-step traceback, debug log to disk or printer, synchronized source file display, breakpoints in source code, disassembly to disk, configurable menus, multiple code and data windows, and keystroke macros.
List: **$395** PC Brand: **$235**

PforCe™

PforCe is a pre-coded optimized object-oriented toolkit of over 400 routines for C programmers. It includes data bases with B-trees, windows, interrupt-driven communications, string handling, menus, all of the basic DOS interfaces, and a complete set of low-level functions to interface directly to the hardware. PforCe comes complete with indexed reference manual, on line resident help, and quick reference card. It supports all memory models of the following C compilers: Lattice, Aztec, Microsoft, CI-86, and Wizard. PforCe includes full source code and there are no royalties on generated applications using the libraries. A demonstration diskette is also available.
List: **$395** PC Brand: **$235**

Pfantasy Pac

A super value pac of Phoenix goodies. Includes Pfix86*plus*, Pmate, Ptel, Plink86*plus*, Pmaker and Pfinish.
List: **$1295** PC Brand: **$875**

For Orders, Literature, or Catalogs, Call Us at...

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

© 1986 PC BRAND Prices, terms, and specifications subject to change without notice.

PC BRAND™: CAREFULLY CHOSEN PROGRAMMER TOOLS

BRIEF *Is Anything But. A Whopper of an Editor*

With a name that belies its thoroughness, Brief™ has every feature you've ever contemplated for your editor-in-chief. Text, from keyboard or files, is housed in multiple buffers, and scrolled through one or more windows you open, close, resize. A text buffer may be called to different windows to view two areas at once. A change in one changes both. Text blocks may be marked for printing, writing to files, movement to scrap buffers for cut and paste into other buffers, or deletion, with as many "undo" levels as you want.

Brief has text search abilities rivaling "grep", with wildcards for matching, indifference to intervening characters, acceptance of character ranges.

If you use Lattice, C86™, or Wizard, and have 320k, you can compile your C program without ever leaving Brief. It finds the lines with errors, and marches you through the text for repairs.

Parts of Brief were written with its own Lisp-like macro language which has structure, 32-character variable names, conditional execution, loops, and you can actually read it! Nothing like the hieroglyphs we've seen elsewhere. Bulletin board and public domain disks with macros. "Simply the best text editor you can buy", Dvorak *Infoworld*. (Needs 192k.)

Ask for:	List:	PC Brand:
U0590	$195	**Call**

HALO GRAPHICS SYSTEM *Multi-Board Graphics Library*

The premier graphics library that got the ball rolling for PC-based graphics and has grown so omnipotent that it supports over 25 graphics boards — including IBM's EGA and Nr. 9 Revolution's hi-res series — and has a multitude of mouse and printer drivers. All that in each box. Separate C versions for Lattice, M'soft, Aztec, CI86. What does Multi-Halo do? A down to the last pixel graphics library plus functions to reset drivers so distributed program can run on anything. Wonderful value for single license. Costly royalties though for redistribution. Specify: S0315 & Language. List: **$300.** We: **$219.** With Dr. Halo II, a free-standing "paint": List: **$440,** Us: **$299.**

WINDOWS for C/WINDOWS for DATA *Microsoft Windows™ and TopView™ Compatible*

Windows for C™ is a library of over 80 functions to add the pizazz and practicality of window partitioning to your application. Unlimited windows, each defined in a C structure for easy reference throughout your program, can be made either to pop up or permanently overwrite the screen. Routines will scroll and highlight lists with arrow keys, will read and scroll ASCII files vertically and horizontally in windows, and even write to memory-loaded files off the screen.

Logical treatment of video attributes permits unchanged programs to run on color or monochrome. Colors of windows are set individually.

All functions are in separate modules; only those used are linked. Only buffers holding on-screen or temporarily obscured windows occupy RAM; others released dynamically. Best overall rating and fastest display in Bill Hunt's 7/85 *Tech Journal* review of five windowing products.

Windows for Data comprises all of Windows for C but takes in data through the windows as well. At the high level a single function lets you specify prompt string, field length, data type, screen location, picture, target variable, then sets lesser functions scurrying to get and process a user's input. There are utilities to get system date and time, mess with strings, create your own masks for fields.

Field options can require entry, prevent entry, permit insert or overtype, beeping on invalid or overflow keystrokes, and attachment of field-specific help messages and functions you want called to display messages or validate entries. And you decide which keys will clear a field, jump to the next or prior, quit, etc. Options diverse enough that a set of "fields" can be made to behave like a Lotus™ menu.

Specify Compiler:	List:	PC Brand:
T0100 Windows for C	$195	$149
T0150 Windows for Data	$295	$259

MICROSOFT C 4.0 *A Great C Battle Rages and You're Winning*

As the dreadnaughts pound each other with ever heavier ordnance, today's programmers reap the spoils of this war. Bundling a source debugger and a "make", and sporting a "huge" memory model permitting single data objects larger than 64k, the Microsoft C compiler has jumped a full version number to 4.0. But what's really impressive are the benchmarks reported in Dr. Dobb's (8/86) encyclopaedic survey of 17 C compilers. Microsoft's and IBM's C (licensed from Microsoft) run away with the contest winning 11 of 27 benchmarks.

The CodeView™ debugger, free for a limited time, uses windows to show everything on one screen: source alongside disassembled object, variables, stack and registers. Drop down windows—use a mouse if you like—obviate learning of commands. "A source-level debugger that puts the rest to shame" (Dobb's),

Microsoft C now has five memory models for code and data, plus non-library support for another thirteen, and boasts alternate math packages for speed versus accuracy, with or without 8087/80287 chips. A big plus in multi-language settings: call from this C any routine written in later versions of M'soft Pascal, FORTRAN, or Macro Assembler. Object code of all four may be intermixed come link time or commingled into libraries.

Both linker and library manager are part of the package, as is the "make", a UNIX™ name for a smart batch program which knows to expend minimum effort to rebuild any size of project by compiling and assembling only elements affected by new or changed modules.

It is reportedly used by Lotus, Ashton-Tate and, fittingly, Microsoft itself to develop Windows. Dobb's calls it "the best MS-DOS C development environment value today [for] virtually any kind of program conceivable." 320k suggested.

Ask for:	List:	PC Brand:
G0500	$450	$295

C-TREE *B-Tree File Manager, Source Code, No Royalties!*

C-tree is sturdy code that has weathered many seasons of prolonged and widespread use. It comes in C source, so you can modify it to fit a special case. No royalties provided you bind it into your binary application.

C-tree's design splits nodes to allow any number of users to access an index file simultaneously even when updates are in progress. So multi-user configurations and adaptation to networks are possible. Record-locking routines are provided for DOS 3.1/3.2, UNIX and XENIX.

Thanks to source code which does not deviate from the K&R standard, C-tree can travel. Tests in many environments prove that C-tree gives your application a ticket to anywhere.

C-tree permits any number of keys for a data file, supports duplicate keys, alphanumeric or numeric, supports files of variable record length; multiple keys in one index file, and keys of variable length. Both high level ISAM routines which handle details with minimum coding, and decomposed step-by-step functions you can access directly. It's comprehensive.

Ask for:	List:	PC Brand:
F0660	$395	$329

CURSES *Unix Style Screen Management*

Curses from Lattice™ manages the screen of the PC like Unix™ curses. Library of 84 functions and macros parallels Unix with matching parameter lists. So Unix programs are at home on the PC, and vice versa. Keeps any number of screens in memory, supports color, vast function set to get characters, wrap lines, scroll, blank lines, highlight, etc. Like Unix refreshes screen only on your command. Ask for: L0850. List: **$125**. Here: **$99**. With Source: L0860, **$250/$199**

dBC *Lattice Library Maintains dBASE Compatible Files With the Power and Speed of C*

dBC™ links C to dBASE. It creates and maintains files and their indexes which exactly replicate dBASE file design. So dBASE can read and update them. And the reverse. dBC can use any files created by dBASE. Now C and dBASE can operate on the same data bases interchangeably.

That opens up the widespread culture of dBASE installations to exploitation by C programmers. Tap that market, avoid the resident dBASE language, and gain the advantages of C with this single product.

dBC's functions parallel all dBASE's file handling commands, many decomposed to give closer control. Each backed by demo source files on disk.

Use dBC for custom work for clients, or on its own. It's a complete ISAM file manager for C whether or not dBASE will be used in tandem, supports all four memory models, and can have sixteen index and data files open. Big discount to buyers of both dBASE II and III versions. Specify Lattice, Microsoft 3.x, or DeSmet.

Versions:		List:	PC Brand:
L00II	For dBASE II	$250	$195
LCCII	With Source	$500	$390
L0III	For dBASE III	$250	$195
LCIII	With Source	$500	$390

PANEL *Feature-Laden Screen Design Tool*

Writing your own screenware can blow completion dates and profits. Panel™ works with you interactively to set up foolproof screen displays and data entry forms rapidly. Output is C source code.

Not just single plane: layer your screen designs with up to ten overlapping images: Background pop-up lists, help boxes, and alternate input fields.

Panel builds in a user interface for keystroke movement within and between fields, supplies validation routines for checking user field entries. Diverse attributes may be selected for any field — size, data type, color, conversion of input to upper case; clearance of existing data when new entry is started; masks for standard formats (eg, dates); phrases which fill in when their first letter is typed; multiple-choice lists from which to choose by cursoring a highlighted bar. Fields may be multi-lined and scrolled if larger than the screen space allotted them. Specify: S0400 & Compiler. List: **$295**, Us: **$229**

30-DAY MONEY-BACK GUARANTEE

We refund the purchase price of any product returned within 30 days in entirely resalable condition. **You can even try out programs themselves** if product code begins with E, T, or L through N — even if it means breaking the disk seal. Some developers do pose limits, so for products beginning with other letters, opening sealed disks constitutes acceptance. But you can at least review the manual. There's just nothing stopping your buying from PC Brand.

WHY US?

- Latest versions of all products
- Everything listed is in stock
- Shipped 24 hours or sooner
- No surcharge for credit card or COD purchase. **NEED TERMS?** On-the-spot credit to most public companies, government, educational, medical institutions. **LOOKING FOR SOMETHING?** We can get many more products — just ask! **NEED MORE INFO?** Our Catalog and literature cover just about everything.

For Orders, Literature, or Catalogs, Call Us at...

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600
PC Brand, 150 5th Ave., New York, N.Y. 10011-4311
Telex: 667962 (SOFT COMM NYK)

© 1986 PC BRAND
Prices, terms, and specifications subject to change without notice.

TODAY's TOP QUALITY AIDS TO PROGRAMMING PRODUCTIVITY

GREENLEAF FUNCTIONS *Bountiful Harvest*

C source, assembler source, and binary libraries of 225 functions for many compilers. Emphasizes tight functional groupings to minimize loading code which your application may never use. Manual helps select functions, bulletin board, too.

A sampling: *DOS* extensions for file and directory manipulation; *Screen:* to select mode, page, monochrome or color, palette; cursor shape, positioning; clearing and scrolling; pixel get and put; read light pen. *Strings:* Center, justify, etc.; efficient list operations which add, delete, sort string pointers for top speed. *Other:* graphics character primitives, keyboard status, function key assignment, time/date, read registers and memory size, peek and poke. Mature best-seller. Specify: S0770 & Compiler. List: $185, Here: $139

GREENLEAF COMMUNICATIONS *Hello World*

Want your application to communicate with other users or remote date bases by asynchronous communications built right into your C programs! Even if you don't need it now, that's a skill to have at the ready!

120 functions *and demo programs in both C and assembler source code* set up separate transmit and receive ring buffers for up to 16 simultaneous channels. Interrupt driven so you can halt an incoming record, display it, file it, let the user edit it, then continue. Goodbye separate communications software.

Supports up to 9600 baud, ASCII or binary, any parity or word length, 8250 UARTs, Xon/Xoff and Xmodem, WideTrack receive. Specify: S0750 & Compiler. List: $185, Us: $139

INTERACTIVE-C

Compiler-Compatible Interpreter, Editor, Debugger

Earlier C interpreters were miraculous compromises. Interactive-C shows how far C interpreters have come. More than an interpreter, Interactive-C is a fully-integrated development environment: a complete K&R interpreter bound tightly to its own editor and debugger.

Slice through programming projects like a hot knife through butter. Extensive error-checking insures immediate detection of program misbehavior. State of the art debugging tools include breakpoints, watchvalues, several stepping options and interactive viewing and modification of variables. An Interactive-C exclusive lets you interrupt to edit and "continue" from where you left off. Eliminates plodding replays of already debugged code—the ball and chain of other interpreters.

Operate Interactive-C using adjustable edit, command, and status windows. Toggle a second screen showing only your program's output—never any crowded intermixing. Or, boost productivity with twin CRTs. Load object code of functions you have already compiled. Or of commercial libraries. Interactive-C has immediate mode, syntax checking both as you type and run, and cursor positioning precisely pointing at an error, not possible with incremental or pseudo-compilers which leave source code behind.

100% compiler compatible—right down to header files and library calls. Port programs between Interactive-C and your compiler with no modifications whatever— not even tricky areas of dynamic memory allocation and I/O.

Specify:	List:	PC Brand:
E950 & Compiler	$249	$219

DAN BRICKLIN'S DEMO PROGRAM

Storyboard Your Program

The Legendary One has created Metaphor Two when the rest of us are still on Zero. Dan's first was the original electronic spreadsheet (VisiCalc™). This one is for programmers.

Words don't express program ideas because programs are screens! Dan's Demo creates slide shows. Create a screen — a snapshot of your planned product as it runs. Anything goes: words, borders, box rules, inverse and underlining of monochrome, fore- and background color. Copy this "slide" to an empty screen. Change it a little, to show the next instant of run-time. Do it again. Presto, a whole slide show of your program in action.

All 250 characters and attributes are available from scrollable lists which pop to the screen. All commands are layered in Lotus-style pop-up menus. Frequent choices mapped to function keys as well. 80x25 character mode, not bit-mapped.

Screen areas can be blocked for cut and paste or filled with color or characters, even blink. Slides can overlay on others, can be shuffled, deleted. Slides can proceed at time intervals or branch anywhere in the slide sequence depending on user keyhits.

Invaluable to prototype the program you are about to write, to position the labels, choose the color decor, smoothe out the keystroke interface. Or load the "capture" utility and snapshot the screens of any running program for an instant slide show.

Each copy entitles you to redistribute fifty of the slide projector program that runs demos. Plain manual, no binder keeps price of big product small. "Might . . . become *the* essential tool in . . . user interface prototyping," *Tech Journal.* Ask for: N0100. List: $75 US: $69

BASTOC OPTIMIZES!

Translates BASIC Into C

For a trifling price, BASTOC™ moves truckloads of BASIC code over to C. It's a translator which takes in Microsoft Extended BASIC and emits pure K&R C for Lattice 3.0. It will optionally convert your program into a single monolithic C function or decompose it into separate functions, one for each GOSUB label.

Version 2's optimization dramatically reduces execution time. Converts to integers those variables in BASIC programs which do not need floating point. Where BASIC uses full assignment statements to increment counters, BASTOC converts to C's compact form. Strings dynamically allocated ridding your application of BASIC's catatonic halts for garbage collection. Creates structure of even convoluted BASIC code. Huge worksaver.

Ask for:	List:	PC Brand:
S0375	$495	$399

Shopping List for the Power Workbench

	LIST	US
ASSEMBLERS & DEBUGGERS		
Advanced Trace-86 *Morgan, ASM Interpreter*	175	119
Codesmith-86 *Dubugger by Visual Age*	145	99
CSD Debugger *C source level by Mark Williams*	75	55
C-Sprite Debugger *by Lattice, source level*	175	139
Microsoft Macro Assembler *with Utilities*	150	109
PASM86 *by Phoenix, Macro Assembler*	195	125
Periscope I Debugger *Data Base Decisions*	295	235
Periscope II *Data Base Decisions*	129	99
Periscope II-X *software only*	115	74
Pfix86 Plus *by Phoenix, Symbolic Debugger*	395	235
BASIC LANGUAGE		
BetterBASIC *Summit Software*	195	165
BetterBASIC Utilities *8087 Math Support*	99	85
Btrieve Interface	99	85
Run-Time Module	250	225
Microsoft BASIC Interpreter *for XENIX*	350	295
Microsoft QuickBASIC Compiler *full BASICA*	99	79
Professional BASIC *by Morgan*	99	69
True BASIC *True BASIC Inc*	150	99
Run Time Module	150	99
True BASIC Libraries *Btrieve, Asyn, Sort, etc.*	Var	Call
C COMPILERS		
C-86 Compiler *Computer Innovations*	395	289
Lattice C Compiler *from Lattice*	500	299
Let's C Compiler *by Mark Williams*	75	55
with CSD Source Level Debugger	150	105
MWC-86: Mark Williams C Development	495	369
Microsoft C Compiler *4.0*	450	295
C INTERPRETERS		
C-Terp *by Gimpel Software*	300	249
Instant C *by Rational Systems*	500	395
Interactive-C *by IMPACC with debugging*	249	219
RUN/C Professional *from Lifeboat*	250	185
RUN/C *without Loadable Libraries*	120	109
TEXT EDITORS		
Brief *from Solution Systems*	195	Call
Edix *by Emerging Tech...Multi-screen*	195	159
Epsilon *by Lugaru Software, like EMACS*	195	149
FirsTime *by Spruce Technology, C syntax*	295	229
Kedit *by Mansfield, similar to Xedit*	125	99
LSE, the Lattice Screen Editor *Multi Window*	125	100
Pmate *by Phoenix, with Macros*	195	115
Text Management Utilities *Grep, splat, diff, etc.*	120	100
Vedit *by Compuview*	150	99
Vedit Plus *by Compuview*	185	129
FILE MANAGERS		
Btrieve *by Softcraft, no royalties*	250	195
Btrieve Network *by Softcraft*	595	465
C-Tree *by FairCom - no royalties, source*	395	329
R-Tree *by FairCom-Report Generator*	295	245
C-Tree & R-Tree Combo *by FairCom*	650	541
dBC *dBASE file manager from Lattice*	250	195
with source	500	390
dbVista *single user DBMS by Raima*	195	139
with source	495	399
dbVista *multi-user DBMS*	495	399
with source	990	815
Opt-Tech Sort *Can sort Btrieve files*	149	105
SCREEN DESIGN		
Curses *by Lattice, UNIX screen designer*	125	99
with Source	250	199
Greenleaf Data Windows.....*New*	225	169
with source	395	297
source purchased later	225	169
On-Line Help *from Opt-Tech Data*	149	105
Panel *by Roundhill, no royalties*	295	229
View Manager for C *by Blaise*	275	189
Vitamin C *by Creative Programming*	150	129
Windows for C *Vermont Creative Software*	195	149
Windows for Data *includes Windows for C*	295	259
ZView *Data Management Consultants*	245	175
GRAPHICS		
Essential Graphics *by Essential, no royalties*	250	210
GSS Graphics Development Toolkit	495	375
GSS Kernel System *by Graphic Software*	495	375
GSS Kernel System *for IBM RT*	795	645
GSS Metafile Interpreter	295	235
GSS Plotting System	495	375
Halo *by Media Cybernetics*	300	219
with **Dr. Halo II**	440	299
Halo for Microsoft *includes all fonts*	595	434
COMMUNICATIONS		
Asynch Manager *by Blaise, for C or Pascal*	175	125
Greenleaf Communications *by Greenleaf*	185	139
PTel *by Phoenix, Binary File Communicator*	195	115
Software Horizons Pack 3	149	119
UTILITY LIBRARIES		
Blaise C Tools Plus	175	125
Blaise C Tools	125	89
Blaise C Tools 2	100	69
C Food Smorgasbord *by Lattice*	150	109
C Utility Library *by Essential, 300 functions*	185	139
Greenleaf Functions *by Greenleaf Software*	185	139
PforCe *by Phoenix, vast library*	395	235
Software Horizons *Packages*	Var	Call
TopView Tool Basket *by Lattice, source avail*	250	199
DEVELOPMENT TOOLS		
Code Sifter *by David Smith Software, Profiler*	119	89
C-Worthy *by Custom Design Software*	295	269
C-Worthy for Network *Menus, help, errors*	495	449
Dan Bricklin's Demo Program *Prototyper*	75	69
LMK from Lattice *by Lattice, "make" like UNIX*	195	149
Microsoft Window Development Toolkit	500	365
PC-Lint *by Gimpel Software, after UNIX's "lint"*	139	125
PFinish *by Phoenix, EXE performance analyzer*	395	235
Plink86 Plus *Utilizes memory for overlays*	495	325
Pmaker *by Phoenix, like UNIX "make"*	125	85
Pre-C *by Phoenix, UNIX "lint"-alike*	295	155
Pfantasy Pac *six Phoenix products*	1295	875
OTHER TOOLS		
BASTOC *by JMI, convert BASIC to C*	495	399
BASIC-C *BASIC's functions added to C*	175	139
The HAMMER *by OES Systems*	195	139
Report Option *by Softcraft, Btrieve Report Gen.*	145	128
Xtrieve *by Softcraft, Query Utility for Btrieve*	245	220
FORTRAN COMPILERS & UTILITIES		
ACS Time Series *by Alpha Computer Service*	495	405
Forlib- Plus *by Alpha Computer Service*	70	45
Microsoft FORTRAN *Links with Microsoft C*	350	219
Microsoft FORTRAN *for XENIX*	695	546
RM/FORTRAN *by Ryan McFarland*	595	Call
Scientific Subroutine Package *by Alpha*	295	239
The Statistician *by Alpha Computer*	295	239
Strings & Things *by Alpha Computer*	70	45
OTHER LANGUAGES & UTILITIES		
Microsoft COBOL Compiler	700	499
Microsoft COBOL Compiler *for XENIX*	995	795
Microsoft COBOL Tools *with Source Debugger*	350	259
Microsoft COBOL Tools *for XENIX*	450	333
Microsoft Lisp *New Common Lisp*	250	189
Microsoft MuMath *includes MuSimp*	300	199
Microsoft Pascal Compiler *Links with M'soft C.*	300	199
Microsoft Pascal Compiler *for XENIX*	695	546
PDisk *Phoenix's new disk manager*	195	125
RM/COBOL *by Ryan-McFarland*	950	Call
RM/COBOL 8X *ANSI 85 COBOL*	1250	Call
Source Print *Aldebaran's diagrammer*	139	109

PRICED TO SAVE YOU MONEY, SHIPPED FAST ANYWHERE.

BEST PRICES YET!

RYAN-McFARLAND FORTRAN

NEW!

A Mighty Fortress Is Their FORTRAN

Picking over features of rival products is not necessary if FORTRAN is your need, still the citadel of scientific and engineering work. Ryan-McFarland has left the competition battering at the gates.

RM/FORTRAN™ is a complete implementation of FORTRAN-77 (ANSI X3.9-1978), the only PC FORTRAN certified by the General Services Administration at the highest test level. The reason: it's a big mainframe compiler moved to PCs, with the bonus that mainframe and mini applications can wander between environments.

Now, on your PC, you can develop large applications, with programs up to 640k (bigger using overlays), arrays over 64k, and using a long list of VS, VAX and FORTRAN-66 extensions you may have grown fond of — long symbolic names, "include", IRT bit functions — because R-M has left out nothing.

But what really sets RM/FORTRAN apart is optimization. The compiler reduces the number of instructions to the minimum which will actually execute, and even takes advantage of each processor's features to deliver lightning-fast object code. It runs 30%-40% faster than Microsoft 3.2, and could make your mainframe not worth the trouble.

Comes with an interactive symbolic debugger like that accompanying IBM VS FORTRAN, Plink86 subset, has a cross reference compile option, supports assembler and C subroutine calls, IEEE floating point, 8087 and 80287 chips.

"Compiler's documentation, ease of use, speed of execution, and debugger facilities place it first for recommendation" said the *Tech Journal* (10/85).

R-M has been writing FORTRAN compilers for IBM, DEC, etc. for 20 years. There is no greater expert.

Ask for:	List:	PC Brand:
I0300	$595	Call

RUN/C PRO

C Interpreter Links Binary Libraries

Run/C comes in an apprentice and pro version. The professional model dynamically loads and unloads multiple binary function libraries like C-Food Smorgasbord™ and Halo Graphics™ — potentially any library compiled with Lattice's large model. Inside this interpreter your C program can reach for functions in the best of commercial libraries.

This C interpreter behaves like PC BASIC meets WordStar® . Use fullscreen editing to create a program. RUN it. If it stumbles, LIST it, EDIT it, RUN it again, fix it again. Use familiar commands like LOAD, MERGE, SAVE, FILES, even TRON and TRACE.

Ideal for program development. Put up code at high speed, try out things devil-may care, let RUN/C find your malaprops. Blast away until tight little code segments are undyingly faithful.

Manual shows how to develop the interface to a commercial library, using the Lattice compiler (a must!). Link your own function archive the same way. (320k minimum; 512k recommended to fit libraries.)

Ask for: S0950 List: $250 PCB: $185

ZVIEW

Screen Design Aid

A complete package for screen design with full windows management as a bonus! Easy creation of screens with complex validation, such as range checking or required/optional data. Powerful Screen Paint utility for creating or editing applications screens. Built in security levels, set at run-time, control read or read/write access by field or screen. Automatic help screen processing for run-time aid per field or screen. Applications regain control during field tabbing, allowing run-time on-screen transaction processing or flow control. Run-time functions include Screen Read and Write with automatic transparent data conversion from screen image to data storage, Field Editing, Help Screen Processing, even a capability to change any field characteristic at run-time, plus Window Push Pop and Scroll. Versions for Lattice, Microsoft and Aztec C. Automatic free updates to registered users. No run-time royalties.

List: $245 PC Brand: $175

LATTICE C COMPILER

Major Upgrades to the Best Selling C Compiler

Lattice now embraces key UNIX™ enhancements which have entered the language since K&R: void functions returning no value, enumerated data types to assign stepped values to variables, data passing between structures by assignment.

The greatly expanded libraries (325 functions!) enable the file sharing and record locking provisions of DOS 3.1, provide a full complement of transcendentals, and a host of utilities to mimic the UNIX and XENIX™ environments.

Lattice 3.0 defaults to the ANSI proposed standard when you need strict adherence, but command line options restore leniency. And it adopts ANSI checking of external function arguments by data type to kill bug swarms when modules join up at link time.

Lattice now delivers smaller .EXE files, boasts very fast link times and a more efficient aliasing algorithm. New options generate code to use 80186 and 80286 features; 8087 of course sensed and utilized. Lattice has enjoyed pre-eminence so long that developers have created far more snap-on tools for Lattice C than any other compiler. William Hunt's *PC Tech Journal* review of 12 compilers awarded Lattice the only "very good" rating for add-on library availability.

Ask for:	List:	PC Brand:
S0100	$500	$299

FOREIGN POLICY

We ship anywhere. Phone or Telex your order. Credit cards: We need card number, expiry date, name and address of card. Or wire funds to PC BRAND, c/o Chemical Bank, 126 East 86th St., NY NY 10028, Account No: 034-016058. **We'll ship immediately** and confirm by Telex if you provide number.

BETTER BASIC

Convert Microsoft BASIC. Structured, Compilable.

Combines the familiarity of BASIC with the best features of C, Pascal, and Modula 2, yet BetterBASIC is 100% compatible with Microsoft's GW™ BASIC and IBM BASICA including graphics, sound, and assembly language calls. So load your old programs and RUN. SAVE and they are converted automatically to BetterBASIC!

It's big: Needs 192k; programs can go to the PC's full 640k. It's comfy: Behaves like M'soft BASIC at the interactive level, with a full-screen editor, direct statement execution, and always poised to RUN. It's fast: Each statement checked and compiled once, not every time encountered. Sieve runs *6 times faster* than with M'soft.

C-like structures house file records so goodbye to FIELD, MKI$, CVD, LSET, etc. Named "procedures" replace GOSUBs to linenumbers. Lots more features: built-in linker for compiled modules; trace; debugging breakpoints; cross-reference command; 32k strings; DOS and BIOS calls and interrupts; recursion. Run-time module stores object code for redistribution.

Ask for::	List:	Us:
S1200 BetterBASIC	$195	$165
S1201 Run-time Module	$250	$225
S1202 8087 Interface	$ 99	$ 85
S1205 Btrieve Interface	$ 99	$ 85

GSS GRAPHICS SYSTEM

Leave the Device Driving to GSS

ANSI CGI STANDARD! PRICES CUT!

GSS™ has reconfigured two components of its comprehensive graphics tools to conform with the ANSI Computer Graphics Interface (CGI) standard.

At the heart of the system is the Development Toolkit which contains all language interfaces and device drivers for keyboards, mice, joysticks, tablets, printers, plotters, cameras, and more. Drivers house management of vector graphics (plotters) and bitmaps used by raster input devices (scanners) to insulate the application program from concern for device idiosyncracy. No one else has implemented CGI that way. It means your programming remains generic; just switch drivers and the same program will drive a different device.

GSS Kernel™ conforms to level 2b of ANSI's Graphical Kernel System (GKS) and contains all its needed drivers and language bindings. Kernel has macro level tools to draw and color an object, store the sequential instructions, and recreate the object on its own, as well as segment it, transform it, etc. So powerful, a single command may represent several score lower level statements.

Plotting has the equivalent GKS tools for graph and chart generation and their captioning: hand it apples and oranges, say "pie", and it bakes the numbers into a digestible display for screen or plotters.

Kernel and Plotting have tools to convert images they create to ANSI Computer Graphics Metafiles (CGMs), a tokenized standard for storing every form of graphic image as data. The Metafile Interpreter reads the contents of a CGM and interprets it with full CGI capability for re-creation on various devices.

Quality software? IBM thinks so. They sell the GSS series under their own label.

Unit royalties and annual fees have been instituted for redistribution. Needs 256k.

Ask for:	List:	PC Brand:
GS010 CGI Dvlpmt Toolkit	$495	$375
GS020 Kernel System	$495	$375
GS025 Kernel for IBM RT	$795	$645
GS030 Plotting System	$495	$375
GS040 Metafile Interpreter	$295	$235

BTRIEVE

ASK ABOUT XTRIEVE & RTRIEVE

Queen B-tree File Manager Abdicates Royalties

There's no longer a tithe to incorporate Btrieve™ in applications, a welcome proclamation if royalties would ruin your profit margins. Btrieve takes complete charge of all file creation, indexing, reading, writing, insertion, deletion, space recapture, forward and backward searching. It builds function call "commands" right into the language you use: interfaces to C, Pascal, BASIC, and COBOL, with sample programs in all four, come with each copy.

Btrieve has mainframe specifications! Its balanced-tree indexing scheme finds any key in a million in four or less accesses. Files may have up to 24 indexes; fixed record length to 4090 characters; indexes up to 255 characters; files of 4 billion bytes. Can even extend a file across two drives — even two hard disks!

Version 4.x speeds DOS interaction for large multiply-keyed files; enables variable length records of virtually any length; verifies accuracy (optionally) with read after write, useful in gritty environments; offers password and data encryption.

There's also Xtrieve, for Btrieve file inquiry and data manipulation, and Rtrieve for report writing. All three in versions for any network that supports the MS-DOS 3.1 file sharing function.

Ask for:	List:	PC Brand:
S0650	$250	$195
S0652 Network Version	$595	$465

TERMS AND CONDITIONS OF SALE

Licenses: Each price is for a license to use a product on a single computer and does not constitute its ownership. We will inquire for you about site licenses. Except as otherwise indicated or where "#" follows the Product Code, products may be used to create programs for distribution without royalty payments or additional licenses, provided said programs do not substantially replicate the products themselves.

Compatibility: PC BRAND's standard products are designed to operate with the IBM® PC, XT or AT under PC-DOS and require no more than 128k of RAM unless indicated. Non IBM machines using MS-DOS: contact manufacturer about precise differences so we can advise.

Returns: See box page one. Defective parts will be replaced. Please call for authorization to return a product for refund.

Payment: We honor MasterCard, Visa, American Express (no surcharge), checks in advance, or funds wired to PC Brand, c/o Chemical Bank, 126 East 86 St., New York, Account 034-016058. COD (U.S. only) for cash, money order, certified check (no fee). NY State, add sales tax. Purchase orders accepted from larger corporations and institutions at our discretion if you agree to net 30 days plus 2% a month late penalty thereafter.

Shipping & Handling: U.S.: UPS Surface: 1st product $6, each add'l $3. UPS 2nd Day Air: 1st product $10, each add'l $4.50. UPS Next Day Air or Federal Express 1-2 Day Air: 1st product $18, each add'l $6. FedEx Next Day 10 AM: 1st product $28, each add'l $7. International: Charges vary by destination and carrier. $10 per shipping container for export forms. Air parcel post at your risk beyond collected insurable amount.

For Orders, Literature, or Catalogs, Call Us at...

800 PC-BRAND

That's (800) 722-7263. In NY State call (212) 242-3600

PC Brand, 150 5th Ave., New York, N.Y. 10011-4311

Telex: 667962 (SOFT COMM NYK)

© 1986 PC BRAND

Prices, terms, and specifications subject to change without notice.



Binary Transfer

RONALD FLORENCE

An XMODEM module for a UNIX-based system allows transmission of data, including non-ASCII files, between a DOS and a UNIX system.



The rapid proliferation of microcomputers has created a PC culture that, to a great extent, has trouble communicating with the older mainframe/minicomputer culture. Many mainframes and some minicomputers have hardware limitations that restrict asynchronous communications to de facto six- or seven-bit words and a limited set of control characters (line feed, carriage return, backspace). By contrast, most PCs make no limitation on the use of all eight bits of each character. Each culture has evolved its own data transmission protocols, and only a few of these protocols have the capability of translating between the two.

In the world of PCs, the XMODEM protocol has become the lingua franca of machine-to-machine transfers. Although many businesses use proprietary transfers such as Microstuf Crosstalk or Hayes Smartcom, which, by using long data blocks of 512 bytes, are efficient and fast, these programs require both the receiver and sender to be set up for the same protocol. For bulletin boards, CompuServe, and direct machine-to-machine links, XMODEM is the norm. Most PC communications programs, whether commercial or public domain, include it. The sidebar "The XMODEM Process," on page 147, briefly explains the protocol. (See also "Screenspeak," Augie Hansen, November 1984, p. 151, for a discussion of Smartcom and XMODEM, and "High-tech Mimicry," Augie Hansen, September 1984, p. 46, for a comparison of 10 communications packages, including Crosstalk.)

Among mainframes and minicomputers, however, XMODEM is virtually unknown, mainly because it uses all eight bits of each transmitted byte. To transfer binary data, the larger machines must use a protocol that includes a quoting mechanism to send eight-bit data in seven bits. The quoting mechanism prefixes a character with a quote character (# or &) to indicate that the high-order bit of the following character is set. The extra characters, including the additional quoting characters (which must be sent to indicate that a # in the data stream is only a # and not a quoting character), exact a high toll in terms of overhead.

Kermit, the protocol developed at Columbia University to bridge mainframe-to-microcomputer transfers, is

ILLUSTRATION • J. A. LAPINSKI

flexible, versatile, and reliable, with the capability of translating file-naming conventions and establishing a packet-exchange handshake between machines. (See "Kermit," Augie Hansen, January 1985, p. 110). Kermit includes quoting mechanisms for machines that cannot send or receive eight-bit data. The price of this flexibility, however, is speed: because Kermit transfers data in short packets of approximately 90 characters, it is somewhat slower than XMODEM. In *any* protocol transfer, each packet includes the overhead of synchronization characters, packet numbers, checksums or CRCs (cyclic redundancy checks), and the wait for the acknowledgment from the remote machine. Hence, the shorter the packets, the greater the overhead. Kermit is available in some PC communications programs, but it is not used extensively.

For many PC users, the most common interaction with minicomputers takes place with systems running UNIX. Frequently, the only file-transfer protocol available between the machines is an ASCII dump. A **stty** command on the UNIX machine can be used to map the line endings from the DOS CR-LF combination to the UNIX NL, or vice versa. To dump a file onto the UNIX system, the remote user enters the command **cat > filename** on the UNIX system. Anything sent from the remote terminal up to an EOF (Ctrl-D for most UNIX systems) is put into **filename** on the UNIX system. To dump an ASCII file from the UNIX machine to a PC logged on as a terminal, the user opens the capture file on the local machine and enters **cat filename** on the UNIX machine. These simple procedures can be enhanced using UNIX commands such as **mesg n**, which prevents other users from writing to the terminal screen during the transfer, or **echo**, which signals the end of the transfer.

These ASCII dumps are efficient, but they offer no error-checking (an increasingly important feature as long-distance carriers compete by cutting line quality in conjunction with costs). Even if the connection is noise-free, a seven-bit ASCII dump cannot transfer binary files. For those who use UNIX machines with cross-compilers as software development systems for DOS, neither object files nor executable programs can be transferred between the UNIX system and a DOS machine using these ASCII dump procedures. For example, a developer using his DOS machine at home with a modem as a terminal on a remote UNIX machine, could not download the executable DOS version of his program from the UNIX machine without an eight-bit protocol.

In addition, UNIX-based machines are limited in their ability to call to other machines by the absence of the XMODEM protocol. The typical university or engineering super microcomputer or minicomputer, running some version of UNIX, communicates with other UNIX machines using **cu** (call up) for interactive communications, or **uucp** (UNIX-to-UNIX communications program) for unattended file transfers. Both programs rely on a streamlined ASCII dump to transfer text or source files. For binary transfers, most UNIX systems use a version of Kermit. These programs work well for communications between UNIX systems, but **cu** and C-Kermit (another such program) lack support for the file-transfer protocols used by most DOS bulletin boards and some large database systems.

UNIX systems have no problem with eight-bit data, as long as the **stty** or **ioctl( )** functions that condition the line are set for eight-bit characters. Thus, XMODEM should be adaptable for use on a UNIX system, so that a caller with a PC can log on using a DOS terminal program, such as Crosstalk or Headlands' PC-TALK, and make error-checked file transfers to and from the UNIX system using XMODEM. (See "PC-TALK: Communications Control," Will Fastie, September/October 1983, p. 162.) For UNIX users with source licenses, the UNIX communications program also can be modified to include XMODEM.

UNIX systems can handle eight-bit data as long as the stty or ioctl( ) functions that condition the line are set for eight-bit characters.

The code in listing 1 (XMODEM.C) is a module, written in C, that will handle the XMODEM transmission or receipt of files to and from a UNIX system, using either checksum or CRC error checking. (See "CRC Calculation," W. David Schwaderer, April 1985, p. 118.) The code includes an option for the transfer of text files between a DOS machine and a UNIX machine; if the text option is invoked, the transfers will make the conversions between DOS CR-LF line-endings and UNIX NLs.

XR AND XT

Compiled with the code in listing 2 (XR.C), the **xmodem** module produces a stand-alone remote XMODEM program for UNIX systems. The compiled program **xr** is linked with another file called **xt**. (A file link in UNIX creates two separate names for the same file.) XR.C parses to determine which name has been used to invoke the program: **xr** is invoked to receive files, **xt** to send files. The executable file(s) should be in a directory on the UNIX system that is on the default path of most users, such as **/usr/bin** or **/usr/lbin**. The invocation format for the two is the same except for the file name, either:

xr [–ct] [–d errfile] filename

or

xt [–ct] [–d errfile] filename

The **–c** option provides CRC error checking instead of the default checksum; **–t** provides text file line-end conversions; and **–d** provides verbose debugging information in a user-specified file on the UNIX system.

To send a file to the UNIX system from the PC using XMODEM, the user would enter the command

xr [options] filename

on the UNIX system. The **xr** program checks to see if it can write the file on the UNIX system or announces that it cannot (if, for example, the user has given a path that has no write permission). If the file already exists, instead of overwriting the file, **xr** will create **filename~**. This is a crude sort of collision-protection scheme, which fails with longer (14-character) file names.

The user then begins the XMODEM file transmission on the PC terminal program. To cancel the transfer, the PC sends a Ctrl-X, (the CAN character used by XMODEM), which is sent automatically by most XMODEM programs when a transfer is aborted.

To send a file from the UNIX system to the PC, the user would enter

xt [options] filename

on the UNIX system. If the file exists, and can be read, **xt** announces that it is ready to begin the transfer and give the size in blocks of 128 bytes. Then, the user would invoke the XMODEM receive procedure on the PC terminal program. The **xt** program automatically adjusts to the error-checking method specified by the receiver program. Again, Ctrl-X aborts the transfer.

If the **–t** option is used to send text files or source code between a DOS

THE XMODEM PROCESS

The XMODEM protocol is prevalent as a method of file transfer in the PC environment. It was originally developed by Ward Christensen for use on machines that were running CP/M operating systems. The protocol's ability to send binary files with some error checking over less-than-perfect telephone lines is the reason for its widespread adoption.

An XMODEM protocol file transfer between two computers is shown schematically in figure 1. The vertical bars show relative time proceeding downward. Various "messages" are depicted as arrows pointing in the direction of the receiver of the message. For the purposes of this discussion, it is assumed that the computer on the left is sending a file to the computer on the right.

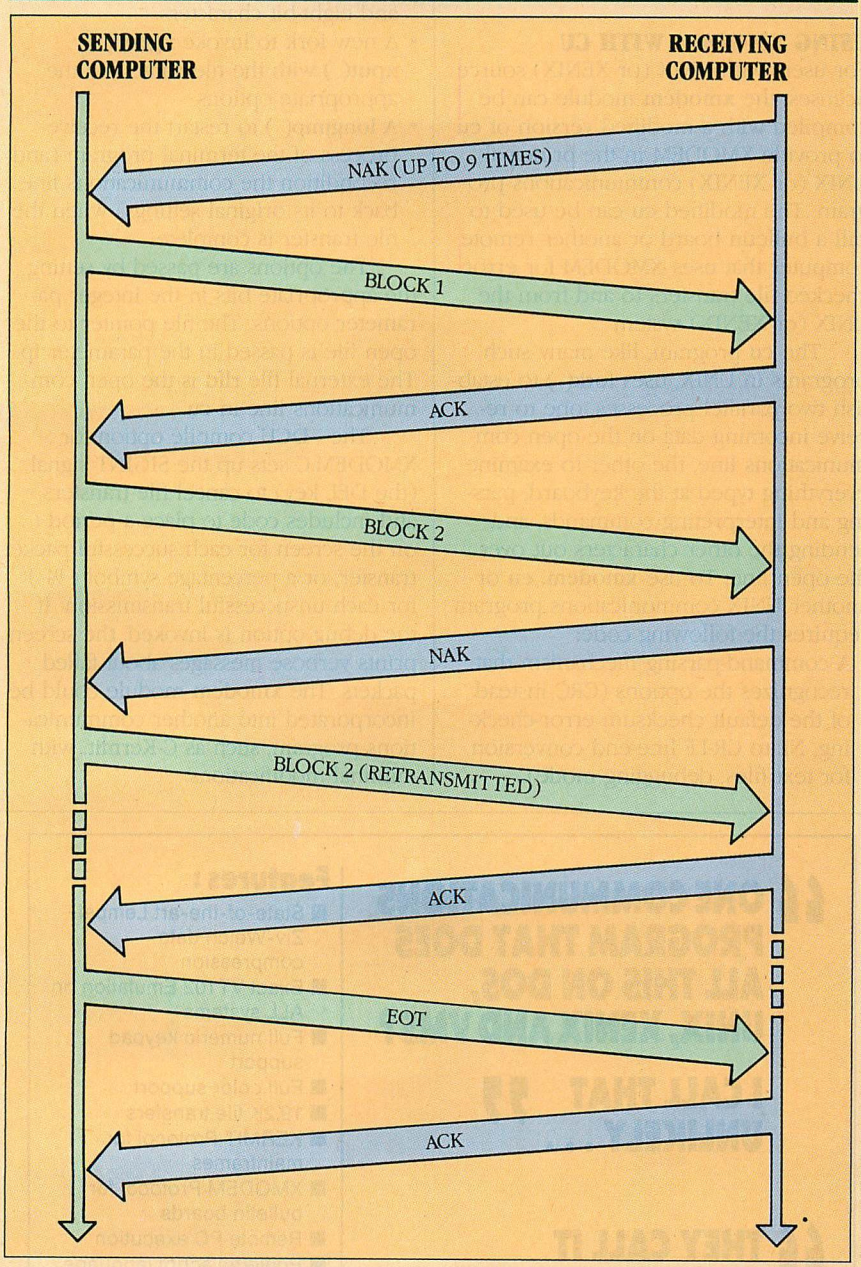
The transfer is initiated when the receiving computer sends an NAK (negative acknowledgment) message to the sending computer, indicating that it is ready to synchronize and receive a file. It will send NAK characters at 10-second intervals until the sender begins transmitting the file or until nine NAKs have been sent, indicating that the sender is not responding.

Assuming that the sender synchronizes, the first block is transmitted. A block, depicted in figure 2, consists of an SOH (start of header) character, a block number (sequential up to 254) expressed as an ASCII character followed by the character equivalent of the one's complement of the block number, then 128 bytes of data, padded if necessary to fill the block, and a checksum value derived from the transmitted data.

The checksum is calculated by adding the ASCII values of the characters in the 128-byte block, then ANDing the results with 255. If the receiver calculates the same checksum value for the received data, an ACK (acknowledgment) is returned to the sender, indicating success. Failure is indicated by an NAK, which would result in a retransmission of the block until it is received correctly or until nine retries have been made. When the sender has no more data to send to the receiver, it transmits an EOT (end of transmission) character and awaits an ACK, which terminates the transfer.

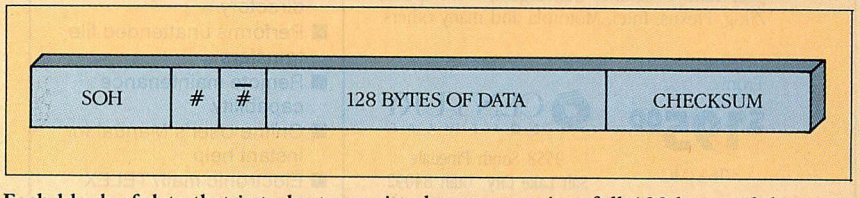
—Augie Hansen

FIGURE 1: *A File Transfer via XMODEM*



The receiving computer (right) will send NAK (negative acknowledgment) characters at 10-second intervals until the sender begins transmitting the file or until nine NAKs have been sent, thus indicating the sender is not responding.

FIGURE 2: *XMODEM Block Description*



Each block of data that is to be transmitted must contain a full 128 bytes. If the block is shorter than 128 bytes, it must be padded to the full length.

system and a UNIX system, the file lengths on the UNIX machine will be shorter than the file lengths on the DOS machine. This is a result of the line-end conversions that take place.

USING XMODEM WITH CU

For users with UNIX (or XENIX) source licenses, the **xmodem** module can be compiled with a modified version of **cu** to provide XMODEM in the principal UNIX (or XENIX) communications program. The modified **cu** can be used to call a bulletin board or another remote computer that uses XMODEM for error-checked file transfers to and from the UNIX (or XENIX) system.

The **cu** program, like many such programs in UNIX, uses **fork( )** to establish two parallel processes, one to receive incoming data on the open communications line, the other to examine everything typed at the keyboard, parsing and interpreting commands, and sending the other characters out over the open line. To use **xmodem**, **cu** or another UNIX communications program requires the following code:

- A command-parsing mechanism that recognizes the options (CRC instead of the default checksum error-checking, NL to CR-LF line-end conversion for text files, debugging mode).
- Code to kill the receive fork of the communications program and recondition the open communications line to no parity, no stripping of characters to seven bits, no XON/XOFF protocol, and eight-bit characters.
- A new fork to invoke **xget( )** or **xput( )** with the file name and the appropriate options.
- A **longjmp( )** to restart the receive process of the terminal program (and recondition the communications line back to its original settings) when the file transfer is complete.

The options are passed by setting the appropriate bits in the integer parameter options. The file pointer to the open file is passed in the parameter **fp**. The external file **rlfd** is the open communications line in **cu**.

The **–DCU** compile option for XMODEM.C sets up the SIGINT signal (the DEL key) to cancel file transfers, and includes code to place a period (.) on the screen for each successful packet transfer, or a percentage symbol (%) for each unsuccessful transmission. If the debug option is invoked, the screen prints verbose messages about failed packets. The **xmodem** module could be incorporated into another communications program, such as C-Kermit, with minimal modifications.

"ONE COMMUNICATIONS PROGRAM THAT DOES ALL THIS ON DOS, UNIX, XENIX AND VMS?

I CALL THAT UNLIKELY..."

"THEY CALL IT TERM"

COMMUNICATIONS SOFTWARE

TERM is available now on AT&T, DEC VAX, MICROVAX, IBM XT, AT, Sun, Altos, HP series 200, 300, Tandy, Fortune, Convergent, NCR, Sperry, Zilog, Plexus, Intel, Motorola and many others.

From $195.00

VISA/MC

CENTURY SOFTWARE

9558 South Pinedale
Salt Lake City, Utah 84092
(801) 531-8512

Features:

- State-of-the-art Lempel-Ziv-Welch data compression
- Exact VT102 Emulation on ALL systems
- Full numeric keypad support
- Full color support
- 19.2K file transfers
- KERMIT Protocol for mainframes
- XMODEM Protocol for bulletin boards
- Remote PC execution
- Powerful script language for customized applications
- Wildcard file send/receive capability
- Auto-login, dial/redial modem control
- Unlimited autodial directory
- Performs unattended file transfers
- Remote maintenance capability
- Online User's Manual for instant help
- Electronic mail/TELEX interface

CIRCLE NO. 188 ON READER SERVICE CARD

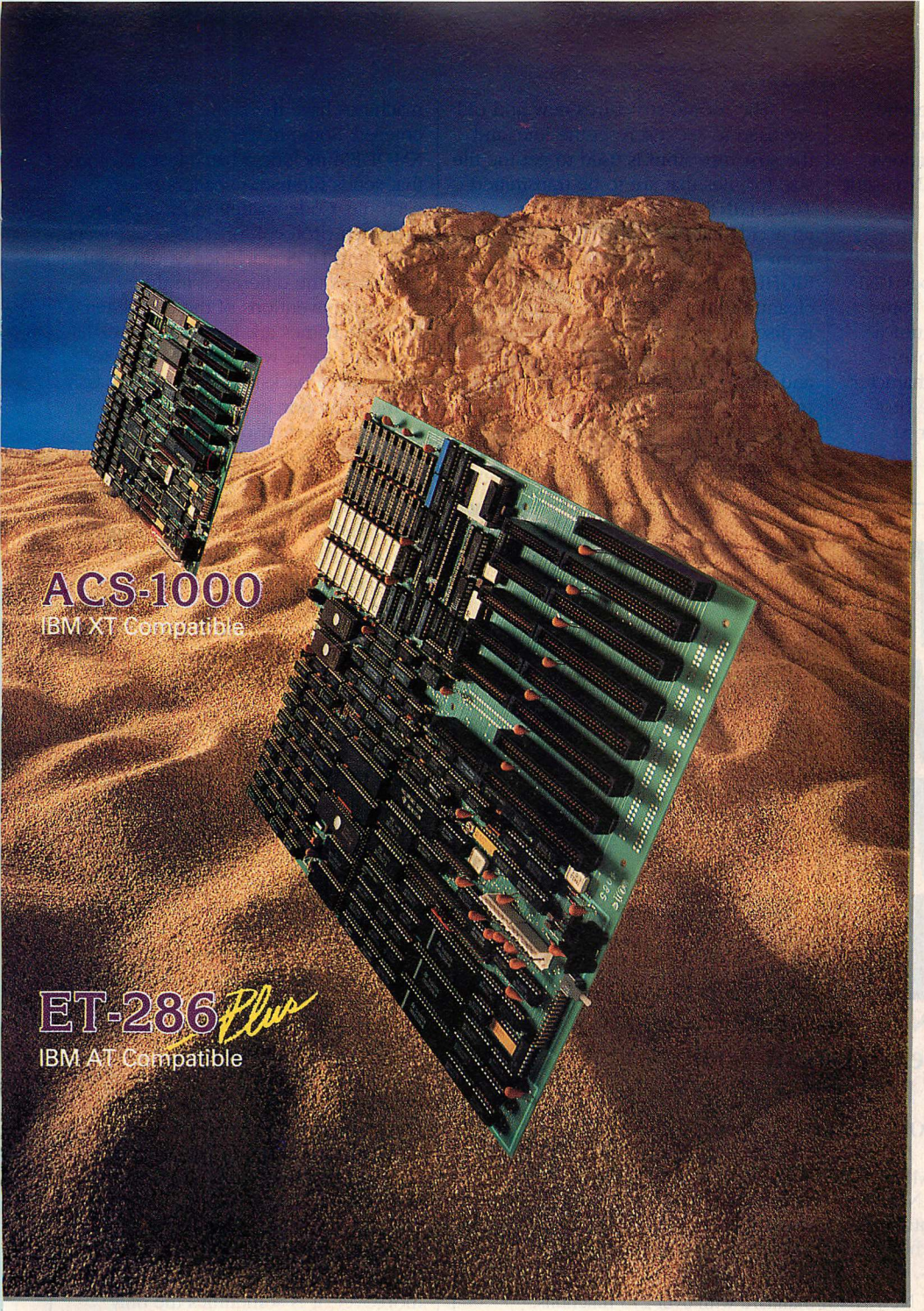
XMODEM.C

The code in XMODEM.C is straightforward. The NOREAD( ) and TX( ) macros and the **err( )** function are used to simplify the code. Static **chars**, rather than **define** statements, are used for SOH (start of header), NAK (negative acknowledgement), ACK (acknowledgement), EOT (end of transmission), CAN (cancel), and **crcinit**, because the UNIX **write( )** call needs the address of the character to be sent. A loop in **xget( )** and the **fillbuf( )** function manage the conversion to and from CR-LF line endings if the text option has been specified. To allow the sender time to set up the XMODEM transmission before the synchronized characters are sent, a **sleep(10)** call is used in **xget( )**. **Sleep( )** is a library function that suspends execution for an interval.

Few constraints are placed on the code for the XMODEM transmission functions; however, the XMODEM receiver must be able to keep up with the sender's transmissions. At 1,200 or even 2,400 bps (bits per second), little danger exists of losing characters; higher transmission speeds place demands on the code. To be certain that characters are not lost, the data bytes are not sent through the checksum or CRC generator until the entire data block is received. Some XMODEM programs available on bulletin boards update screen displays during packet transfers, with the result that they cannot run above 1,200 bps without losing characters.

The bit-level operators in C make the CRC generator a relatively simple function. The bytes that are to be added to the CRC are fed into the generator one bit at a time, high bit first. The bit is shifted into a 16-bit CRC accumulator, and, if the bit shifted out of the CRC accumulator is a 1, an exclusive-or is performed on the CRC accumulator using the bit pattern 1021H. The process repeats for all eight bits of the input character. To make the CRC come out correctly for XMODEM, two zeros must be sent through the generator after the data bytes. These zeros take the place of the CRC bytes in the packet.

The UNIX alarm signal (SIGALRM) is used to signal a time-out on the receive function, by interrupting the system call **read( )** in function **rchar( )**, which returns a –1 to indicate an error. Any other interruption to the **read( )** also causes the read to signal a time-out and the program to recycle and request the block again. Programs that make high-priority system calls to the kernel, such as some windowing programs, interrupt **read( )**, but XMODEM is suffi-



Push Back The Envelope:

10/6, 12 & 16 MHz

EMS Capability

IBM obviously positioned the AT as the hub of the microcomputer network. In this marketplace our products stand out as the pinnacle of performance and value.

SPEED
The ET-286 doubles the clock speed of the AT at 12 MHz and we are already 16 MHz capable. Naturally, the ET-286 toggles from Hyperspeed at 12MHz to 100% 6MHz compatibility for those applications that demand it.

MEMORY
The ET-286 has access to 4 Megabytes of on-board, 0 wait state memory. This memory can be used as regular AT extended memory, or, as Lotus/Intel/Microsoft (LIM) expanded memory (EMS). The ability to utilize the memory on-board as either extended or expanded memory greatly enhances its use in scientific as well as business applications.

COMMUNICATIONS
There are three on board serial ports that are configurable as either RS-422 or RS-232 depending on the application. Two parallel ports further extend your communications ability. And there are still 8 expansion slots for additional I/O.

CUSTOMER SUPPORT
Part of the success of the ACS products is the availability of our people. We provide the support that OEM's need in order to win major contracts: engineering support, competitive pricing specialized packaging, and American made products.

ACS MS-DOS 3.2
We want to help you compete. That's why we have licensed MS-DOS 3.2 and GW Basic 3.2 and made them available in OEM packages at OEM prices.

ACS-1000
- 8 or 4.77 Mhz
- Up to 1 Meg Memory
- 2 Serial Ports
- 1 Parallel Port
- On-Board Floppy Controller
- On-Board Clk/Calendar

ET-286 *plus*
- 10/6, 12 & 16MHz
- Up to 4 Meg Memory (Expanded or Extended)
- 3 Serial Ports
- 2 Parallel Ports
- 8 Expansion Slots
- On-Board Clk/Calendar

NOW AVAILABLE:
- Optional 10Mhz 80287 Coprocessor
- Bundled LIM EMS Driver

CIRCLE 103 ON READER SERVICE CARD

ACS International, Inc.
2105 Luna Rd., Suite 330
Carrollton, Texas 75006

214-247-5151
TELEX: 709748 ACS UD

ACS

IBM, UNIX, XENIX, LOTUS, INTEL and MICROSOFT are trademarks of their respective companies.

ciently robust to immediately resend the packet. Function **cksend( )** uses **rchar( )** to clear the communications line before an ACK, NAK, or CAN is sent back to the XMODEM sender.

XR.C

The code for **xr** and **xt** uses stock UNIX C library functions to parse the command line and to set the communications line for the XMODEM transmission. The initialization for the character **trans???** determines whether the program has been invoked as **xr** or **xt**.

The **termio** structures **new** and **old** are used to set and reset the line, and the structure **stbuf** is used to get the file size for files that are to be transmitted. The SIGINT and SIGQUIT traps prevent an accidental press of DEL or Ctrl-\ from aborting the transmission, and the SIGHUP trap ensures the program's cleanup in the event the telephone line is disconnected accidentally.

The XMODEM protocol is quick and highly accurate for attended file transfers. The code in these modules requires that both sender and receiver machines have the name of the file entered. Some newer versions of XMODEM include a batch-level protocol that sends file-name headers before each file. Code written to incorporate the batch protocol for transfers between UNIX and DOS machines would need to address the difference in the file-naming conventions of the two systems: DOS file names are limited to an eight-character name with a three-character suffix, use a backslash instead of a slash as the separator in path names (except DOS 2.x, which also allows the backslash), and DOS translates file names to uppercase. UNIX permits file names in any combination up to 14 characters; the slash is used as a directory separator; and file names, like commands, are case-sensitive, thus, MYFILE.DOS, myfile.dos, Myfile.dos, and mYfile.dos are all different file names. File names sent from a DOS system are acceptable to a UNIX system, but if they are not mapped to lowercase, they end up at the head of sorted directory listings. (Most UNIX file names are lowercase.) Many UNIX file names, because of their length, format, or the inclusion of characters outside the allowed DOS set, would not be acceptable to DOS machines until they were translated to the format required by DOS.

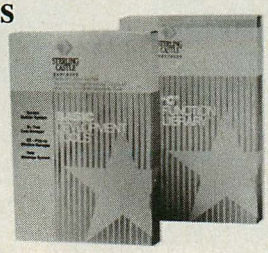
In addition to its uses in transfers between UNIX and DOS machines, XMODEM also can be a time-saver in transfers between UNIX systems. Sending files that have been compressed with **pack** and bundled with **tar** is substantially faster than the ASCII dumps of **cu** and **uucp**. It is particularly convenient to be able to shift from ASCII to binary transfers while using **cu**, instead of having to do some transfers with **cu** and others with C-Kermit.

The XMODEM protocol is a useful bridge between DOS and UNIX. Its transfers are fast, simple, and reliable; moreover, it is sometimes the only means of uploading or downloading binary files or of effecting error-checked transfers of text or source code files between the two systems when a UNIX machine is used to develop DOS programs or when a UNIX machine serves as a terminal for DOS bulletin boards. Indeed, this article and the accompanying code were transmitted from a XENIX system to a DOS machine at *PC Tech Journal* using XMODEM for XENIX and UNIX.

Ronald Florence is a novelist and historian and a devotee of the UNIX operating system. His most recent published work is The Optimum Sailboat *(Harper & Row, 1986).*

HOW TO BECOME A POWER PROGRAMER

Introducing Two New Powerful Packages, BlackStar Basic Development Tools and "C" Function Library



BASIC DEVELOPMENT TOOLS features four-in-one "automatic programing" tools. Including SCREEN BUILDER, B+TREE, HELP MESSAGE SYSTEM and EZ SCREEN POP-UP WINDOWS. Can be used separately or together. Compatible with Microsoft QuickBASIC and Borland Turbo Basic. Comes with 220 page manual and two diskettes.

"C" FUNCTION LIBRARY New ANSI Standard with over 275 functions. A comprehensive development package for the latest "C" Compilers including Microsoft and Lattice. Complete source code and demo program, as well as small, medium and large memory models are included. Plus 317 page instruction manual and three diskettes.

Special Offer! $99.00 each, both for $175.00
For IBM PC and compatibles. M/C, VISA, add $3.00 for shipping and handling. California residents add 6.5%. Shipping outside U.S. and Canada add $15.00.

60 Day Money Back Guarantee.

ORDER TODAY! (800) 722-7853
(213) 306-3020 in California
24-Hour Rush Shipment (credit card orders only).

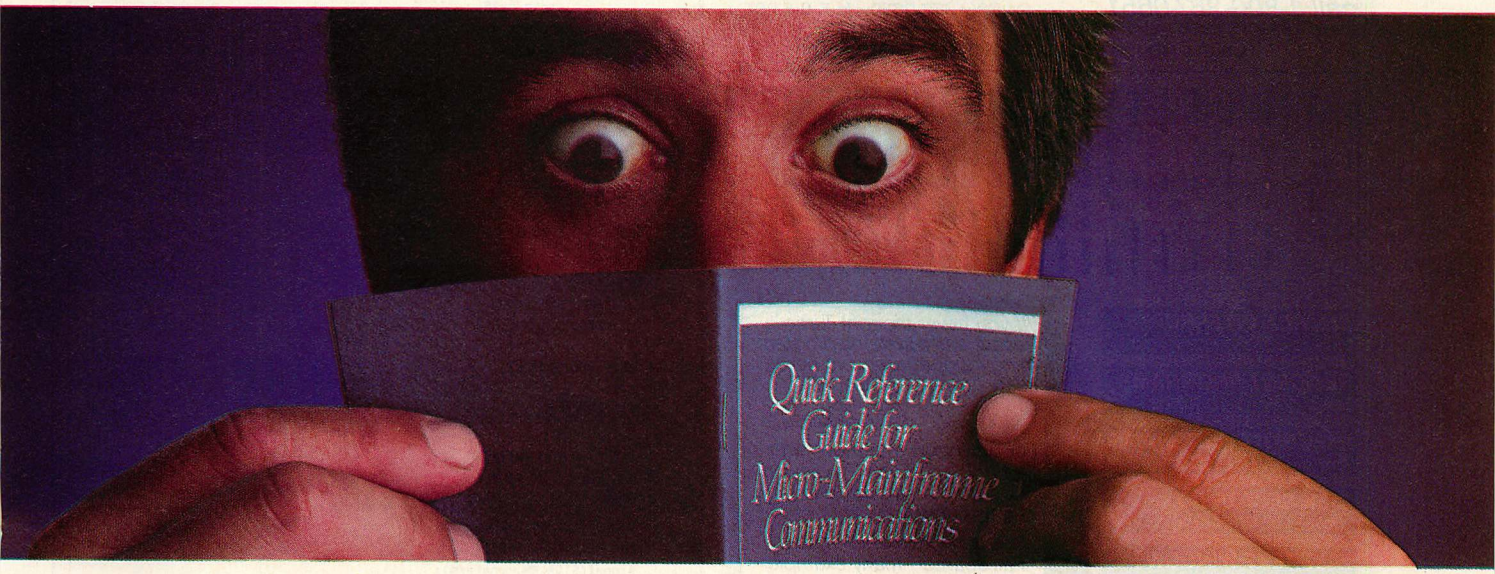
Sterling Castle, 702 Washington Street, Suite 174
Marina del Rey, CA 90292.

STERLING CASTLE™
SOFTWARE

All trademarks acknowledged.

CIRCLE NO. 198 ON READER SERVICE CARD

HERE'S HELP



Attachmate has answers to your questions about micro-mainframe communications. Hardware answers, software answers and now a guide packed full of answers to some important questions—questions worth asking *before* you make long-term decisions.

What should you know about IBM® standards, multiple sessions, windows, file transfer, API, and graphics?

You'll find the answers and solutions in Attachmate's *Quick Reference Guide for Micro-Mainframe Communications*—with a chart comparing IBM, IRMA®, and Attachmate. For a free copy, call toll free:

1-800-426-6283

Attachmate

Micro-Mainframe Technology: We put our heart in it!

Attachmate Corporation
3241 118th S.E., Bellevue, WA 98005
(206) 644-4010

Copyright ©1986, Attachmate Corporation. IRMA is a registered trademark of Digital Communications Associates, Inc. IBM is a registered trademark of International Business Machines Corporation.

CIRCLE NO. 165 ON READER SERVICE CARD

TASKVIEW™

WHY GIVE UP. . .

BATCH FILES,

I/O REDIRECTION

SIDEKICK™

DOS MENU PROGRAMS,

MOST OF YOUR RAM,

EXECUTION SPEED?

Compatible, efficient DOS multi-tasking.

We designed Taskview with efficiency in mind. During normal operation, TASKVIEW hides behind DOS, providing you with control of up to 10 concurrent or non-concurrent programs. Just the touch of a key instantly switches a program to the foreground. Included desktop utilities let you cut and paste from program to program. Simple to use and reasonably priced, no well equipped PC user should be without it.

Requires: PC/AT/Jr compatible, DOS 2.0-3.1, 256K RAM, 1 Floppy drive.

Taskview trademark of Sunnyhill Software
Sidekick registered trademark of Borland Intl.

30-day money back guarantee

Dealer Inquiries Invited.

$79.95 plus $5.00 S&H

Washington residents add 7.9%
International orders add $5.00
VISA and Mastercard accepted.

To order Toll-Free
call 1-800-367-0651

Sunny Hill Software
13732 Midvale N. Ste. 206
Seattle, WA 98133
(206) 367-0650 M-F, 8-6 PDT

CIRCLE NO. 158 ON READER SERVICE CARD

TURBO PROFESSIONAL™

SERVICE INTERRUPTS
No assembly required

RESIDENT PROGRAMS
Easy, pop-up routines

EXECUTIVE PROGRAMS
Run ANY DOS program

DISK SECTOR I/O
Lowest level access

FAST TEXT WINDOWS
Virtual windowing system

KEYBOARD MACROS
Simple, powerful

LOTS OF EXAMPLES
21+ full example programs

MUCH MORE. . .
Over 140 routines in all

"If you never thought Turbo Pascal was a systems programming language, you've never seen Turbo Professional."
Darryl Rubin
Computer Language

For programs that move with technology—Turbo Professional—a truly professional library of subroutines.

150 page reference manual.
Full source—many example programs.
No royalties charged for applications.

Requires IBM compatible, DOS version 2.0 or greater, Turbo Pascal 2.0 or greater.

Turbo Professional, trademark of Sunnyhill Software
Turbo Pascal, registered trademark of Borland International

Dealer Inquiries Invited.

$69.95 plus $5.00 S&H

Washington residents add 7.9%
International orders add $5.00
VISA and Mastercard accepted.

To order Toll-Free
call 1-800-367-0651

Sunny Hill Software
13732 Midvale N. Ste. 206
Seattle, WA 98133
(206) 367-0650 M-F, 8-6 PDT

CIRCLE NO. 152 ON READER SERVICE CARD

LISTING 1: XMODEM.C

```
/* XMODEM.C copyright 1986 Maple Lawn Farm, Inc.
 * exit: 0 if successful, -1 for failure, compile with mlfcu.c:
 *       cc -i -O -s -DCU mlfcu.c XMODEM.C -o cu or with XR.C:
 *       cc -i -O -s XR.C XMODEM.C -o xr, ln xr xt */

#include <signal.h>
#include <stdio.h>

#ifdef CU

extern int      rlfd;           /* the open line in cu */
#define WFD     rlfd
#define RFD     rlfd
#define errf    stderr

#else

extern  FILE    *errf;          /* error file for remote */
#define WFD     1               /* stdout */
#define RFD     0               /* stdin */

#endif

#define BSIZE           128
#define DEBUG           01
#define LF              02
#define CRC             04
#define NOREAD(x, c)    (rchar(x, &c) == -1)
#define TX(c)           write(WFD, &c, 1)
#define ever            (;;)

static  char    soh = 0x01,
                eot = 0x04,
                ack = 0x06,
                nak = 0x15,
                can = 0x18,
                crcinit = 'C',
                cksum;

static  int     debug,
                crc;

int     kleenex(),
        onalarm();

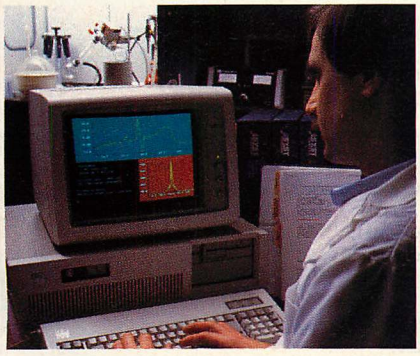
unsigned        crcsum;

xget(fp, opts)
FILE    *fp;
int     opts;
{
        char    buf[BSIZE],
                b = 1,
                inch,
                crchi;
        int     iput = BSIZE;
        register        i;

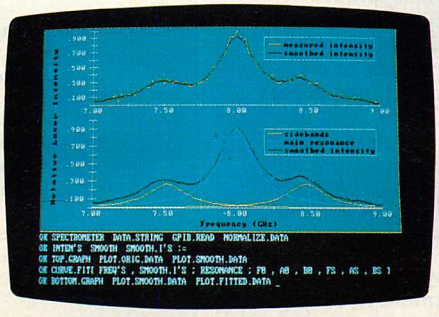
        debug = (opts & DEBUG);
        crc = (opts & CRC);
        signal(SIGALRM, onalarm);
#ifdef CU
        signal(SIGINT, kleenex);
#else
        sleep(10);
#endif
        (crc) ? TX(crcinit) : TX(nak);
        for ever {
            if NOREAD(10, inch) {
                err("Timeout during SOH");
                cksend(crc ? crcinit : nak) ;
                continue;
            }
            if (inch == eot)
                break;
            if (inch == can) {
                err("CAN block %u", b);
                kleenex(-1);
            }
```

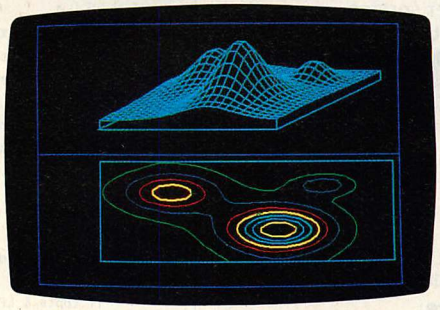
Still the only integrated scientific analysis, graphics, and data acquisition software.

ASYST.™



Macmillan's widely-acclaimed ASYST™ Scientific Software still offers more built-in capabilities than any other PC software. Its unique interactive programming environment includes such functions as FFT, smoothing, integration, differentiation, curve fitting, statistics, differential equations, and matrix and polynomial operations. And all functions are supported by sophisticated graphics, such as error bars, scatter plots, and axonometric plotting.





ASYST's extensive function set provides the building-blocks that let you assemble scientific applications – fast. RS-232 and optional A/D and GPIB/IEEE-488 support offer interface versatility you just can't get from other products. And all hardware support integrates seamlessly with the full range of ASYST capabilities.

Free technical support. 60 days of free technical support help you design, set up, and fine-tune the perfect system. Ongoing support is available at a nominal cost.

30-Day No-Risk Offer.
CALL 1-800-348-0033.

In New York state, (212) 702-3241.

Ask for technical literature, including performance benchmarks.

See Us at Pitcon.
Booths 26034-26036.

Macmillan Software Co.

An affiliate of Macmillan Publishing Company
630 Third Avenue, New York, NY 10017

ASYST is a trademark of Macmillan Software Co. ASYST was developed by Adaptable Laboratories Software, P.O. Box 18448, Rochester, NY 14618

CIRCLE NO. 110 ON READER SERVICE CARD

The ultimate solution for information interchange...

The ultimate 9-track magnetic tape subsystem for the IBM-PC/XT/AT and compatibles!



Innovative Data Technology offers a variety of 1/2 inch 9-track magnetic tape subsystems for the IBM-PC/XT/AT featuring the new "LEO" PC tape controller.

"LEO" is a state-of-the-art PC tape controller that lends itself to the most sophisticated applications including real time data acquisition, multitasking and true streaming disk back-up. "LEO" comes standard with an impressive list of features that include a dedicated microprocessor, up to 64K of RAM for buffering, high speed thruput with memory mapping I/O—no DMA channel required, ASCII to EBCDIC code conversion and external cable connectors for easy installation.

Supplied on a 5 1/4" diskette is the most comprehensive set of software drivers and utilities available. Users have a choice between an installable I/O driver with modules to "Basic" and "C" languages, or an MT-DOS device driver, which allows direct tape access under any language supported by DOS 3.1. Also included is "ANSI," a sophisticated file transfer utility, "TAP" a comprehensive disk back-up and restore utility and "TCMD" tape command, a valuable tool for inspecting tape data and format. All utilities are menu driven with help screens for user friendly operation.

IDT manufactures the complete subsystem ... controller and tape drive. With a commitment to excellence, IDT staffs a complete customer service department, offering you assistance to assure top performance at all times. Contact us today for additional information.

INNOVATIVE DATA TECHNOLOGY

5340 Eastgate Mall • San Diego, CA 92121
(619) 587-0555 • TWX: (910) 335-1610

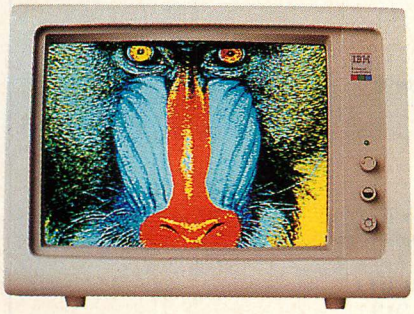
Western Regional Office:
10061 Talbert Ave., Suite 202
Fountain Valley, CA 92078 • (714) 968-8082

Eastern Regional Office:
One Greentree Center, Suite 201
Marlton, NJ 08053
(609) 596-4538 • TWX: (710) 833-9888

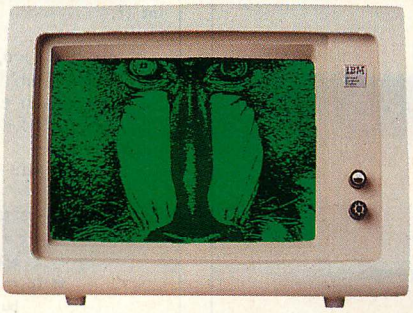

```
        if (inch != soh) {
            err("Bad SOH block %u: %#x", b, (inch & 0xff));
            cksend(nak);
            continue;
        }
        if NOREAD(2, inch) {
            err("Timeout block %u during blocknum", b);
            cksend(nak);
            continue;
        }
        if ((inch & 0xff) != b) {
            err("Expected blocknum %u, got %u", b, (inch & 0xff));
            cksend(nak);
            continue;
        }
        if NOREAD(2, inch) {
            err("Timeout block %u during ~blocknum", b);
            cksend(nak);
            continue;
        }
        if ((inch & 0xff) != (~b & 0xff)) {
            err("Expected ~blocknum %u, got %u",
                    (~b & 0xff), (inch & 0xff));
            cksend(nak);
            continue;
        }
/* Read in 128 byte block without taking time for checksums or crc. */
        for (i = 0; i < BSIZE; i++)
            if NOREAD(2, buf[i])
                break;
        if (i < BSIZE) {
            err("Timeout data recv, char #%d", i);
            cksend(nak);
            continue;
        }
        if (crc) {
            if NOREAD(2, crchi) {
                err("Timeout crc hibyte");
                cksend(nak);
                continue;
            }
            crchi &= 0xff;
        }
        else if NOREAD(2, inch) {
            err("Timeout %s", (crc) ? "crc lobyte" : "checksum");
            cksend(nak);
            continue;
        }
/* Now, when we have the whole packet, do the checksum or crc. */
        for (cksum = 0, crcsum = 0, i = 0; i < BSIZE; i++)
            upsum(buf[i]);
        if (crc) {
            upsum(0);       /* needed for crcsum */
            upsum(0);
            if ((inch & 0xff) + (crchi << 8) != crcsum) {
                err("Expected crc %u, got %u",
                        crcsum, (inch & 0xff) + (crchi << 8));
                cksend(nak);
                continue;
            }
        }
        else {
            cksum %= 256;
            if (cksum != (inch & 0xff)) {
                err("Expected checksum %u, got %u",
                        cksum, (inch & 0xff));
                cksend(nak);
                continue;
            }
        }
        TX(ack);
#ifdef CU
        putc('.', stderr);
#endif
        if (opts & LF)
            for (i=0, iput=0; i < BSIZE; i++) {
                if (buf[i] == 0x1a)  /* old ms-dos eof */
                    break;
                if (buf[i] != '\r')
                    buf[iput++] = buf[i];
            }
```

EGAWONDER™

Any Software. Any Monitor. Any Time.[1]



Any Software on an EGA Monitor



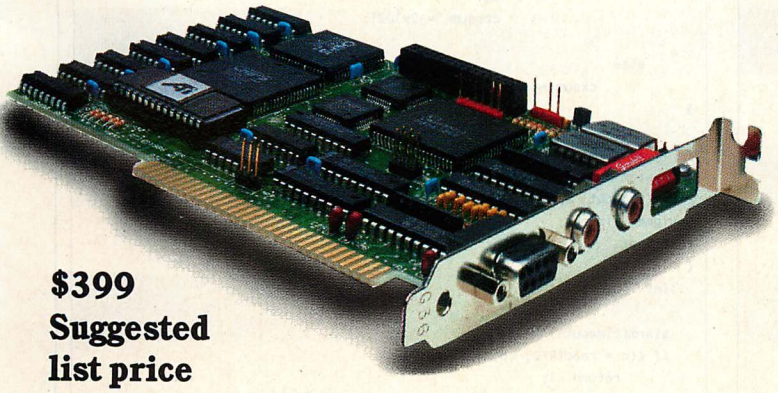
Any Software on a TTL Monochrome Monitor



Any Software on an RGB Color Monitor[3]

Upgrade to EGA without an EGA Monitor
The unique **ATI EGA WONDER** allows upgrade to the new EGA graphics standard without the purchase of an expensive EGA monitor. **EGA WONDER** runs EGA, CGA, MDA, Hercules and 132 column software on EGA Color, RGB Color, TTL Monochrome and Composite monitors. Extremely flexible, **EGA WONDER** maintains downward compatibility to both existing software and existing monitors. On the internal monitor of a Compaq PC Portable, **EGA WONDER** displays EGA, CGA, MDA and Hercules software via an optional expansion module.[2] For EGA monitor users, the **ATI EGA WONDER** improves the display of CGA software by producing high resolution 8×14 text and double scanned graphics. Old CGA software is now displayed with EGA quality. Completely compatible to IBM's EGA, **ATI's EGA WONDER** performs smooth scrolling, pixel panning and windowing. No memory modules are required because 256K of video memory is a standard feature. **EGA WONDER** is the only card able to display EGA software on the internal monitor of both the IBM PC Portable and the Compaq PC Portable. **EGA WONDER** provides an NTSC Composite signal for interface to a Polaroid Palette.

SoftSense Automatic Mode Switching
User friendly and easy to use, the **ATI EGA WONDER** has built-in SoftSense Automatic Mode Switching. It is actually able to sense and automatically switch between EGA and CGA color modes or between EGA, MDA and Hercules monochrome modes. The **ATI EGA WONDER** is compatible at the hardware level to the IBM Enhanced Graphics Adapter, the IBM Color/Graphics Adapter, the IBM Monochrome Display Adapter and the Hercules Graphics Card to minimize software incompatibilities.



$399 Suggested list price

EGA WONDER is now available from all major computer stores, call us today at **(416) 477-8804** for more information.

CIRCLE NO. 206 ON READER SERVICE CARD

ATI® TECHNOLOGIES INC.

Technology you can Trust.

ATI Technologies Inc., 450 Esna Park Dr., Markham, Ontario, Canada L3R 1H5. TLX. 06-966640.

Trademarks IBM PC Portable – EGA CGA MDA – International Business Machines – Compaq – Compaq Computer Corporation – Hercules – Hercules Computer Technology – Polaroid Palette – Polaroid Corporation.

1. Any Software, Any Monitor, Any Time applies to IBM graphics, standards, monitors, software.
2. Optional Compaq Expansion Module (no 132 Columns) Suggested list price $99.
3. EGA, MDA, Hercules software displayed via interlacing. Flickering effect of interlacing reduced with purchase of anti-glare screen filter.

```
        fwrite(buf, iput, 1, fp);
        b++;
        b %= 256;
    }
    TX(ack);
    kleenex(0);
}

xput(fp, opts)
FILE    *fp;
int     opts;
{
    register    i;
    char    buf[BSIZE],
            b = 1,
            cb,
            crclo,
            inch;
    int     cread;

#ifdef CU
    signal(SIGINT, kleenex);
#endif
    signal(SIGALRM, onalarm);
    debug = (opts & DEBUG);
    rchar(60, &cb);
    if (cb == crcinit)
        crc = 1;
    else if (cb == nak)
        crc = 0;
    else {
        err("No startup %s", (crc) ? "'C'" : "NAK");
        kleenex(-1);
    }
    cread = fillbuf(fp, buf, (opts & LF));
    while (cread) {
        for (i = cread; i < BSIZE; i++)
            buf[i] = 0;
        TX(soh);
        TX(b);
    cb = (-b & 0xff);
    TX(cb);
    write(WFD, buf, BSIZE);
    for (cksum = 0, crcsum = 0, i = 0; i < BSIZE; i++)
        upsum (buf[i]);
    if (crc) {
        upsum(0);       /* needed for crcsum */
        upsum(0);
        crclo = crcsum;
        cb = (crcsum >> 8);
        TX(cb);
        TX(crclo);
        }
        else {
            cksum %= 256;
            TX(cksum);
        }
        if NOREAD(15, inch) {
            err("Timeout after block %u", b);
            continue;
        }
        if (inch == can) {
            err("CAN after block %u", b);
            kleenex(-1);
        }
        if (inch != ack) {
            err("Non-ACK after block %u: %#x", b, inch);
            continue;
        }
#ifdef CU
        putc('.', stderr);
#else
        if (debug)
            fprintf(errf, "Validated block %u\n", b);
#endif
        cread = fillbuf(fp, buf, (opts & LF));
        b++;
        b %= 256;
    }
    for ever {
        TX(eot);
        if NOREAD(15, inch) {
            err("Timeout during EOT");
            continue;
        }
        if (inch == can) {
            err("CAN during EOT");
            kleenex(-1);
        }
        if (inch != ack) {
            err("Non-ACK during EOT: %#x", inch);
            continue;
        }
        break;
    }
    kleenex(0);
}

fillbuf(fp, buf, lf)
FILE *fp;
char *buf;
int lf;
{
    int     i = 0, c;
    static  int     cr_held;
    if (cr_held) {
        buf[i] = '\n';
        i++;
        cr_held--;
    }
    for (; i < BSIZE; i++) {
        if ((c = getc(fp)) == EOF)
            break;
        if (c == '\n' && lf) {
            buf[i] = '\r';
            if (i == 127) {
                cr_held++;
                return BSIZE;
                }
                buf[i+1] = '\n';
                i++;
            }
        else
            buf[i] = c;
    }
    return i;
}

upsum(c)
char    c;
{
    register  unsigned  shift;
    register  unsigned  flag;

    if (crc)
        for (shift = 0x80; shift; shift >>= 1)  {
            flag = (crcsum & 0x8000);
            crcsum <<= 1;
            crcsum |= ((shift & c) ? 1 : 0);
            if (flag)
                crcsum ^= 0x1021;
        }
    else
        cksum += c;
}

/* Timeout in rchar() works by deliberately interrupting the read()
 * system call.  errno=EINTR, so no reason for a perror() autopsy. */
rchar(timeout, cp)
unsigned timeout;
char *cp;
{
    int     c;

    alarm(timeout);
    if ((c = read(RFD, cp, 1)) == -1)
        return -1;
    alarm(0);
    return c;
}
```

```
onalarm()
{
   signal(SIGALRM, onalarm);
}

kleenex(sig)
int sig;
{
#ifdef CU
   if (sig > 0)
        cksend(can);
   else
        fprintf(stderr, "\r\nFile transfer %s.",
             (sig) ? "cancelled" : "complete");
   fprintf(stderr, "\r\n");
#else
   printf("File transfer %s.\r\n", (sig) ? "cancelled" : "complete");
   resetline();
#endif
   exit(sig);
}

cksend(ch)
char ch;
{
   int j;
   char cp;

   do {
        j = rchar(2, &cp);
   } while (j != -1);
   TX(ch);
}

/* VARARGS1 */
err(s, i, j)
char    *s;
int     i, j;
{
     if (debug) {
          fprintf(errf, s, i, j);
#ifndef CU
          fprintf(errf, "\n");
     }
#else
          fprintf(errf, "\r\n");
     }
     else
          putc('%', stderr);
#endif
}
```

LISTING 2: XR.C

```
/* XR.C - remote xmodem functions for xenix/unix
 * copyright 1986 Maple Lawn Farm, Inc.
 * usage: xr|xt [-ct] [-d errfile] file
 * -c   crc (instead of checksum), -t   text mode (CR-NL <-> NL)
 * compile:     cc -i -O -s XR.C XMODEM.C -o xr
 *              ln xr xt
 * To avoid overwriting existing files, received file with same name
 * as an existing file is stored as fname~. */

#include <signal.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <termio.h>

#define DEBUG       01
#define LF          02
#define CRC         04
#define BSIZE       128
#define errx(m,f)   printf("%s: ", pname), \
                    printf(m, f), \
                    printf("\n"), \
                    exit(1)

FILE    *errf;

struct  termio  old,
                new;
hangup()
{
        resetline();
        exit(1);
}

main(ac, av)
int     ac;
char    **av;
{
        char    *pname,
                trans = (av[0][strlen(av[0])-1] == 't'),
                *fname;
        int     c, opts = 0;
        struct  stat    stbuf;
        extern  int     optind;
        extern  char    *optarg;
        FILE    *fp;

        pname = *av;
        while ((c = getopt(ac, av, "ctd:?")) != EOF)
          switch (c)  {
          case 't' :
                opts |= LF;
                break;
          case 'c' :
                opts |= CRC;
                break;
          case 'd' :
                opts |= DEBUG;
                if (!(errf = fopen(optarg, "w")))
                    errx("can't open %s", optarg);
                setbuf(errf, NULL);
                break;
          case '?' :
                printf("usage: %s [-ct] [-d errfile] file\n", pname);
                exit(1);
          }
        if (ac == 1 || ac == optind)
          errx("need file name", NULL);
        fname = av[optind];
        if (trans && !(fp = fopen(fname, "r")))
          errx("can't open %s", fname);
        if (!trans) {
          if (!access(fname, 0))
                strcat(fname, "~");
            if (!(fp = fopen(fname, "w")))
                errx("can't write %s", fname);
        }

        printf("Ready to %s %s\n", (trans) ? "send":"receive", fname);
        if (trans) {
          stat(fname, &stbuf);
          printf("%d blocks (128 bytes/block)\n",
                stbuf.st_size/BSIZE+1);
        }
        printf("Ctrl-X to abort transfer\n");

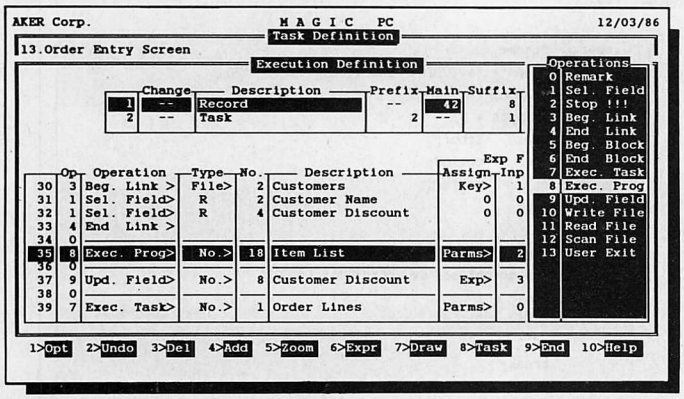
        signal(SIGINT, SIG_IGN);
        signal(SIGQUIT, SIG_IGN);
        signal(SIGHUP, hangup);

        ioctl(1, TCGETA, &old);
        ioctl(1, TCGETA, &new);
        fflush(stdin);
        new.c_iflag = IGNBRK|IGNPAR;
        new.c_oflag = 0;
        new.c_lflag = 0;
        new.c_cc[4] = 1;
        new.c_cflag &= ~PARENB;
        new.c_cflag |= CS8;
        ioctl(1, TCSETAW, &new);

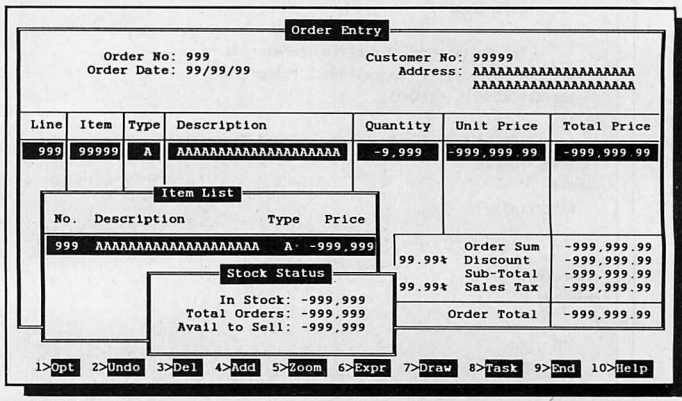
        (trans) ? xput(fp, opts) : xget(fp, opts);
}
resetline()
{
        ioctl(1, TCSETA, &old);
}
```

Announcing Magic PC – the first breakthrough for database applications developers in over 20 years:

Now you can develop professional applications 1000% faster than your 4GL or DBMS, totally free from programming, commands and syntax!



A Magic PC program looks as simple as this. To design an application you quickly fill-in menu-driven decision tables **without having to write a single line of code.** For example, just by highlighting the Execute Program operation on this screen and also highlighting the Item List program in the Program Menu, you tell Magic PC to pop-up the Item List window shown in the adjacent screen, when the end-user hits the Zoom key.



Magic PC gives your end-user the power to harness and retrieve data instantly, **without any commands or syntax** because at runtime you already have built-in options to Add, Delete, Modify, Query and get on-the-spot ad-hoc information simply by highlighting selections from menus. Dats validation, security and error-checking are done automatically for you by Magic PC without programming.

Who needs another DBMS?

At last, Magic PC gives you the ultimate applications design tool, far ahead of 4GL's, DBMS and Application Generators.

Magic PC breaks through the language barrier with the revolutionary Un-Language concept:

NO PROGRAMMING, COMMANDS OR SYNTAX!

Free yourself from your programming language

Magic PC makes you, the professional, completely free from the drudgery of procedural programming. No more cryptic commands, syntax or unforgiving procedural structures, because Magic PC does all the programming automatically. There's your competitive edge. The rest is up to you . . .

The Professional Choice

Already an international success, Magic PC is a profit maker and career booster for DP Consultants, System Integrators, VARs, MIS professionals, System Analysts, Programmer Analysts and Software Engineers. If you design PC applications professionally, you can't afford not to Un-Language now.

IBM France: "IBM encourages this introduction and can not help but salute such evolution . . ."

Israeli Air Force: "We were convinced that it was not possible to have a design tool powerful enough to implement real-life applications without a programming language. Magic PC changed our mind . . ."

Jeff Duntemann, PC Tech Journal: "It's probably the best integrated database applications and screen generator that I have ever seen . . .very smooth system, and smoothness comes at a premium these days . . ."

The Magic PC Secret

You're so much more productive with Magic PC because there is **absolutely no programming** to slow you down. You design a Magic PC application by simply filling-in the **Data Dictionary Tables** (Files, Fields, Keys) and the **Task Description Tables** (Operations and Expressions).

Only 13 design **Operations** harness the power of Magic PC. Operations are specific enough to eliminate the need for tiresome syntax, yet elastic enough to produce robust custom applications. Use the Operations to describe **what** you want and Magic PC makes it happen. It's that simple.

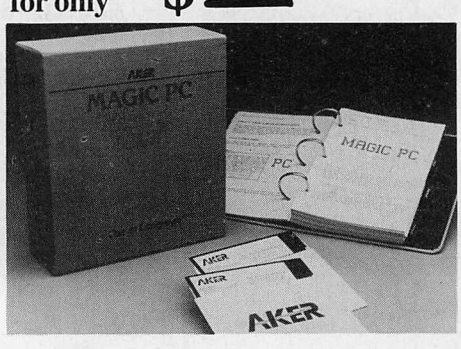
Make Task nesting power available with a single **Execute Task** Operation. This powerful instruction triggers Magic PC to execute and display additional tasks or even external applications through **Window Zooms.** The 3-dimensional effect of Window Zooming lets you probe deep into your application through nested windows and manipulate the data underneath.

You describe a Magic PC Task or Program (composite Tasks) by filling your system analysis flow into the Task Description Tables. Choose the participating Data View, and Magic PC executes your desired Operations. You interface with the Tables by highlighting your selections from pop-up menu-driven windows. There's nothing to edit except your headings.

You're not confined to any particular design sequence as you are with most procedural languages. You can enter and change any Table spontaneously, on the fly, as ideas come to mind and Magic PC automatically maintains the application integrity.

A **Magic Inference Engine** automatically orchestrates your Task Description Tables into a single file of internal **Knowledge Base Rules** for optimum, bug-free performance. Knowledge Base Rules are executed by the **Magic Run** engine for stand-alone runtime operation, or by the **Magic Lan** engine for unrestricted Novell network sharing. You're free to design the Knowledge Base without worrying about the internal structure.

Discover fast, language-free programming at no risk for only $19.95



See for yourself how fast you can program language-free applications with our low-cost limited offer.

You'll get the full Magic PC software unprotected and limited to 100 records and 450 page documentation complete with a **free** Order Entry sample application. You'll also get our **free** telephone support for 90 days!

And your $19.95 will be credited towards the full $695 Magic PC purchase price. Even if you don't buy Magic PC right away, keep your $19.95 Magic PC Trial as your application prototyping tool at this bargain price.

Our No-Risk Guarantee!

You have our no-risk 30-day money-back guarantee: if you're not completely satisfied for any reason, even Magic PC Trial for $19.95, send it back for a refund.

Order now while supply lasts

Call this toll free number now with your Visa, MasterCard or American Express for immediate delivery, or send the Order Coupon below today to Aker

1-800-345-MAGIC
in CA call 714-250-1718

VISA MasterCard AMERICAN EXPRESS

Yes, please rush me:

☐ Magic PC Trial	$ 19.95
☐ Magic PC	$695.00
Add shipping	$ 5.00
In CA 6% tax	$_____
Prices valid in US only.	Total $_____

Ship to:_____

Address:_____

City/ST/Zip:_____

Phone:_____

AKER

Aker Corp. 18007 Skypark Circle B2, Irvine, CA 92714
(714) 250-1718, Elec. Mail Dialcom 41:AKR 001 Telex 4931184
AKR UI OEM and VAR inquiries are welcome.
CIRCLE NO. 153 ON READER SERVICE CARD

Min. requirements PC DOS 2.0, IBM PC or 100% compatible with 512K and hard disk

©1986 Aker Corp. Printed 1/87 **Trademarks:** Magic PC, Un-Language, Window Zoom Magic Run, Magic LAN and Magic PC Trial are trademarks of Aker Corp., IBM P and PC-DOS are trademarks of IBM Corp., Novell is a trademark of Novell Inc.

Procedural Algorithms in Prolog

Strategies are presented for expressing algorithms in Prolog as well as translating algorithms from other languages.

Effective programming is aided by the use of reference books that are complete, and most Prolog textbooks do a thorough job of explaining how to express *information* in Prolog. For example, **bird(tweety).** means "Tweety is a bird," and **fly(X) :- bird(X).** means "X flies if X is a bird."

Much less has been written about how to express *algorithms* in Prolog. Often, a computation is best described not as a set of properties and relations, but as a procedure—perhaps one that already has been developed and debugged in a conventional language.

This article presents strategies for translating algorithms from other languages into Prolog. The examples are in standard Edinburgh Prolog and will run in most implementations without modification. Except where noted, the techniques are equally applicable to Borland's Turbo Prolog, though the sample programs will require changes, such as adding declarations and changing I/O statements. (For a review of three Prolog compilers, see "Prolog Arrives," Michael Covington and Andre Vellino, November 1986, p. 52. See also "Programming in Logic," Michael Covington, December 1985, p. 82 and January 1986, p. 145; and "Programming for AI," Expert Consultant: *Applied AI*, Richard L. Schwartz and Robert E. Shostak, December 1986, p. 191.)

Prolog is often described as a nonprocedural language. It is really a semiprocedural language; a compromise between procedural and nonprocedural programming, it yields some advantages of each. In a truly nonprocedural language, the programmer would specify only a logically rigorous set of conditions the program must fulfill; the computer would then automatically generate an algorithm from them. Such factors as the order in which the conditions were written thus have no effect.

In the interest of efficiency, Prolog contains some procedural elements.

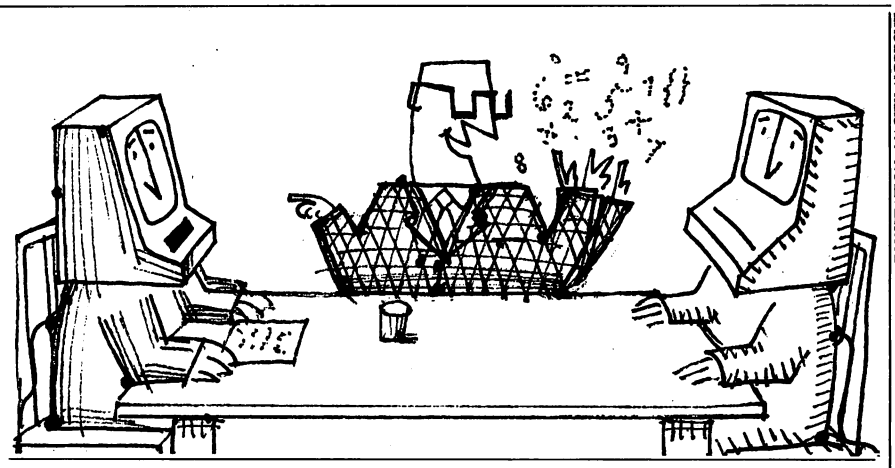


ILLUSTRATION • MACIEK ALBRECHT

The Prolog programmer specifies not only the rules and facts to be used in solving a problem, but also the order in which they are to be attempted. Crucially, the programmer can even specify that some potential paths to a solution should not be tried at all. This makes it possible to carry out computations that would be severely inefficient, or even impossible, in pure Prolog.

The key principle of Prolog is the *procedural interpretation of logic*. Consider the following Prolog rule set:

```
in_north_america(X) :- in_usa(X).
in_usa(X) :- in_georgia(X).
in_georgia(atlanta).
```

This can be interpreted as a set of facts

```
X is in North America if X is in the USA.
X is in the USA if X is in Georgia.
Atlanta is in Georgia.
```

or as a set of procedure definitions

```
To prove that X is in North America,
    prove that X is in the USA.
To prove that X is in the USA,
    prove that X is in Georgia.
To prove that Atlanta is in Georgia,
    do nothing.
```

The unfamiliar task of drawing inferences from data is thereby reduced to the familiar task of calling procedures.

Prolog predicate definitions will be referred to throughout this article as *procedures*, and goals will be referred to as *procedure calls*.

CONDITIONAL EXECUTION

A very important difference between Prolog and other programming languages is that, in general, Prolog procedures have multiple definitions (or clauses), each being applied under different conditions. In Prolog, conditional execution is expressed, not with **if** or **case** statements, but with these alternative definitions of procedures.

Consider for example how the following Pascal procedure might be translated into Prolog:

```
procedure writename(X:integer);
begin
  case X of
    1: write('One');
    2: write('Two');
    3: write('Three')
  end
end;
```

This tanslation could be done by giving **writename** three definitions:

```
writename(1) :- write('One').
writename(2) :- write('Two').
writename(3) :- write('Three').
```

Each definition matches in exactly one of the three cases. A common mistake is to write the clauses as follows:

```
/* poor style */
writename(X) :- X=1, write('One').
writename(X) :- X=2, write('Two').
writename(X) :- X=3, write('Three').
```

This gives correct results but wastes time. Because X is a variable, each clause matches every procedure call; so the computer tries each clause, gets part way into it, and then backs out if X has the wrong value. It is faster to design the clauses so that only the correct one can be invoked in the first place.

A key to effective programming in Prolog is to make each logical unit into a separate procedure. Each **if** or **case** statement should, in general, become a procedure call. For example, the hypothetical Pascal procedure:

```
procedure a(X:integer);
begin
  b;
  if X=0 then c else d;
  e
end;
```

should go into Prolog as follows:

```
a(X) :- b,
        c_or_d(X),
        e.
c_or_d(0) :- c.
c_or_d(X) :- X<>0, d.
```

This imposes a disciplined organization that is even more rigorous than the structured ("goto-less") programming style that underlies Pascal, C, and Ada.

Consider another version of **writename** that includes a catch-all clause to deal with numbers the names of which are not given:

```
procedure writename(X:integer);
begin
  case X of
    1: write('One');
    2: write('Two');
    3: write('Three')
  else
    write('Out of range')
  end
end;
```

(Standard Pascal does not allow the use of **else** with the **case** statement, but most popular compilers do.) One way to express the same algorithm in Prolog is the following:

```
writename(1) :- write('One').
writename(2) :- write('Two').
writename(3) :- write('Three').
writename(X) :- X<1, write('Out of range').
writename(X) :- X>3, write('Out of range').
```

This gives correct results but lacks conciseness. In order to make sure that only one clause can be executed with each number, the value of X was tested in each of the last two clauses. In order to tell the program to print "Out of range" for any number that has not matched any of the first three clauses, the following clauses could be tried, with limited success:

```
/* incorrect */
writename(1) :- write('One').
writename(2) :- write('Two').
writename(3) :- write('Three').
writename(_) :- write('Out of range').
```

(Recall that the anonymous variable, written as _ , matches anything.) The problem here is that the goal **writename(1)**, for example, matches both the first clause *and* the last clause. If a subsequent goal fails and causes backtracking through this one, the goal **writename(1)** will have two solutions: one that prints "One" and one that prints "Out of range."

Writename should be *deterministic*—that is, it should give exactly one solution for any given set of parameters and not give alternative solutions upon backtracking. Therefore, if any of the first three clauses succeeds, the computer should be told not to try the last clause. This can be done with the *cut* operator (written as !).

The cut operator commits the computer to take a particular (or potential) solution without trying alternatives. Suppose that **b** is defined as follows:

```
b :- c, d, !, e, f.
b :- g, h.
```

and that the current goal is **b**. If the cut is executed on the first clause, then it becomes impossible to look for alternative solutions to **c** and **d** (the goals that precede the cut in the same clause) or to **b** (the goal that invoked the clause containing the cut). It remains possible, of course, to backtrack all the way past **b**—outside the scope of the cut—and look for alternatives to the clause that caused **b** to be invoked.

What is needed to do in the incorrect **writename** expression is to put a cut in each of the first three clauses. This changes their meaning slightly, so that the first clause (for example) says, "If the parameter is 1, then write 'One' and do not try any other clauses."

```
writename(1) :- !, write('One').
writename(2) :- !, write('Two').
writename(3) :- !, write('Three').
writename(_) :- write('Out of range').
```

Because **write** is deterministic, it does not matter whether the cut is written before or after the call to **write**. Programs are usually more readable, however, if cuts are made early.

GUARANTEEING THE OUTCOME

In order to control the flow of program execution, it is often necessary to guarantee that a goal will succeed regardless of the results of the computation that it performs. Occasionally, it is necessary to guarantee that a goal will always fail.

An easy way to make any procedure succeed is to add a clause to it that succeeds with any parameters and is attempted last:

```
f(X,Y) :- X<Y, !, write('X less than Y').
f(_,_).
```

A call to **f** succeeds with any parameters; it may or may not print its message, but it certainly will not fail and, hence, will not cause backtracking in the procedure that invoked it. Moreover, because of the cut, **f** is deterministic (provided its parameters are already instantiated). The cut prevents the second clause from being used to generate a second solution with parameters that have succeeded with the first clause.

Similarly, a procedure can be guaranteed to fail by adding cut and fail at the end of each of its definitions:

```
g(X,Y) :- X<Y, write('X less than Y'), !, fail.
g(X,Y) :- Y<X, write('Y less than X'), !, fail.
```

Any call to **g** ultimately fails for one of two reasons: either it does not match any of the clauses present, or it matches one of the clauses and ends with cut and fail. The cut is written next to last so that it will not be executed unless all the other steps of the clause have succeeded first; as a result, it is still possible to backtrack from one clause of **g** to the other as long as the cut has not yet been reached.

In Prolog implementations that allow goals to be treated as data (and this does not include Turbo), the procedures **make_succeed** and **make_fail**, which make any goal succeed or fail, can be defined as:

```
make_succeed(Goal) :- call(Goal), !.
make_succeed(_).

make_fail(Goal) :- call(Goal), !, fail.
```

In some implementations, **call(Goal)** is written simply as **Goal**.

Likewise, the procedure **once** can be defined so that it allows a goal to succeed exactly once, thus making any goal a deterministic one:

```
once(Goal) :- call(Goal), !.
```

This procedure backtracks as much as necessary to get one successful solution to **Goal**, then stops. Thus, no matter how many possible solutions f(X) has, the goal **once(f(X))** returns only the first solution. If f(X) has no solutions, **once(f(X))** fails.

REPETITIVE COMPUTATION

Prolog offers two ways to perform computations repetitively: backtracking and recursion. Of the two, recursion is far more versatile. However, backtracking has some interesting uses, such as the construction of **repeat-fail** loops. In Prolog implementations that lack tail-recursion optimization, **repeat-fail** looping is the only kind of iteration that can be performed ad infinitum without causing a stack overflow.

The predicate **repeat** is built into most Prolog implementations. If not built in, it can be defined as follows:

```
repeat.
repeat :- repeat.
```

(The built-in version should be used if available, because in some implementations, the definition above does not prevent stack overflow.)

The **repeat** predicate always succeeds and has an infinite number of solutions. Thus, any procedure call bracketed between **repeat** and **fail** will be tried over and over again, even if it generates only one solution. For example, the following goal displays an infinite number of asterisks:

```
repeat, write('*'), fail.
```

The following procedure turns the computer into a typewriter, accepting characters from the keyboard and displaying them ad infinitum, until the the Break key is used to abort:

```
typewriter :- repeat,
              get0(C),
              put(C),
              fail.
```

The loop can be made to terminate by allowing it to succeed eventually, so that backtracking stops. The following version of **typewriter** stops when a line terminator (ASCII code 10) is typed:

```
typewriter :- repeat,
              get0(C),
              put(C),
              C = 10.
```

If C is equal to 10, execution terminates; otherwise, execution backtracks to **repeat** and proceeds forward again through **get0(C)** and **put(C)**.

The looping in the latter version of **typewriter** can be restarted by the failure of a subsequent goal (as in the compound goal **typewriter,fail**). To prevent the loop from restarting, a cut needs to be added as follows:

```
typewriter :- repeat,
              get0(C),
              put(C),
              C = 10,
              !.
```

In effect, this forbids looking for alternative solutions to **typewriter** once one solution has succeeded.

A crucial difference does exist between **repeat-fail** loops in Prolog and **repeat-until** loops in Pascal. In Pascal, iteration is accomplished by first executing all the statements in the loop, then jumping from the end back to the beginning. In Prolog, however, backtracking may cause control to jump backward from *any* goal to *any* earlier goal that has alternative solutions. (The limiting case is **repeat**, which does have alternative solutions.) If any goal in a Prolog loop fails, subsequent goals are not attempted.

A serious limitation of **repeat-fail** loops is that information cannot be conveniently passed from one iteration to the next. Prolog variables lose their values upon backtracking. Thus, there is no easy way to make a **repeat-fail** loop accumulate a count or total. (Information can be preserved by storing it in the knowledge base by using **assert** and **retract**, but this process is usually slow and awkward.) With recursion, information can be transmitted from one pass to the next through the parameter list. This is the main reason for preferring recursion as a looping mechanism.

RECURSION

Recursion is a familiar means of implementing task-within-a-task algorithms, such as tree searching and Quicksort. Prolog lends itself well to expressing recursive algorithms developed in LISP. However, *any* iterative algorithm can be expressed recursively.

Here is the classic recursive algorithm for computing factorials, as expressed in Pascal

```
function factorial(N:integer):integer;
begin
  if N=0 then
    factorial:=1
  else
    factorial:=N*factorial(N-1);
end;
```

and as expressed in Prolog (change is to = in Turbo Prolog)

Personalize your computing environment.

The MKS Toolkit now contains the Korn shell command interpreter.

The MKS version of Bell Labs' Korn shell has this and more:

- the full power of the UNIX System V.2 Bourne shell
- the most requested features of Berkeley's C shell
- the full-UNIX utility of executable shell files
- command aliases
- interactive command-line facilities
- previous command history and editing
- a powerful programming language
- shell variable expansion
- arithmetic evaluation

All this has been fine-tuned to create the optimum environment under DOS. The Korn shell is just one of over 100 commands — fully compatible with UNIX System V.2 — now contained in the MKS Toolkit, including the following:

awk	cat	chmod	cmp	cp	cpio	ctags	cut	date
dd	df	diff	du	echo	ed	egrep	ex	fgrep
file	find	head	help	join	lc	ls	more	mv
nm	od	paste	pg	prof	rm	sed	size	sort
split	strings	tail	time	touch	tr	uniq	vi	wc

and much, much more . . .

These programs run from the shell or command.com under DOS on machines such as the IBM PC, XT, and AT, the AT&T 6300, and most PC compatibles. Full documentation is included. Phone support is available 9-6 EST. Not copy protected.

Everything for only $139.

Mortice Kern Systems Inc.
43 Bridgeport Road East, Waterloo, Ontario, Canada N2J 2J4
For information or ordering call collect: **(519) 884-2251**

Prices quoted in U.S. funds. MasterCard and VISA orders accepted. OEM and dealer inquiries invited. UNIX is a trademark of Bell Labs. MS-DOS is a trademark of Microsoft Corp.

CIRCLE NO. 174 ON READER SERVICE CARD

Microstat®

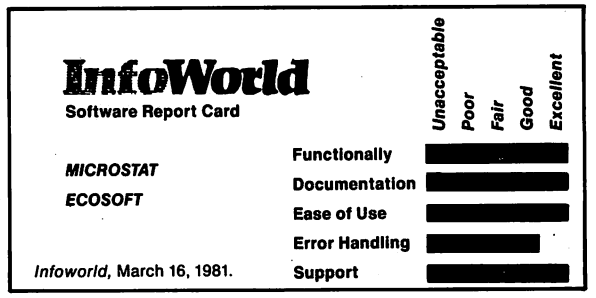
HIGH POWER WITHOUT THE HIGH PRICE

Microstat® has been the most popular statistics package for microcomputers since we introduced it in 1978. In the past two years, Microstat has been requested by name on more military contracts than any other statistics package. When it comes to coverage, ease of use, accuracy, and value, Microstat is unbeatable. Just some of it features include:

- Data Management Subsystem for file creation and management.
- Data Transformations
- Hypothesis Testing
- Three types of ANOVA
- Simple, Multiple, Stepwise Multiple Regression
- 11 Nonparametric Tests
- Factorials, Permutations, Combinations
- Batch or Interactive Operation
- Read external files (e.g., Lotus, dBaseII, ASCII)
- Descriptive Statistics
- Scatterplots
- Correlation Analysis
- Time Series
- 8 Probability Distributions
- Crosstabs and Chi-Square
- User's Manual

Microstat® is available for MSDOS, PCDOS, CP/M80, CP/M86. The price is $375.00. Multiple copy discounts and cost-effective site licenses are available.

To order, call:
800-952-0472
(for orders)
or
317-255-6476
(tech. info.)



Ecosoft Inc.
6413 N. College Ave.
Indianapolis, IN 46220

MasterCard VISA

CIRCLE NO. 131 ON READER SERVICE CARD

Personal REXX
for the IBM PC

★ Interpreter for the full REXX language, including all of the standard REXX instructions, operators, and built-in functions

★ Sophisticated string manipulation capabilities

★ Unlimited precision arithmetic

★ Direct execution of DOS commands from REXX programs

★ Built-in functions for DOS file I/O, directory access, screen and keyboard communication, and many other PC services

★ Compatible with VM/CMS version of REXX

★ Uses include:

— Command programming language for DOS

— Macro language for the KEDIT text editor

— Can be interfaced by application developers with other DOS applications, written in almost any language

Mansfield Software Group, Inc.
P. O. Box 532
Storrs, CT 06268
(203) 429-8402

$125 plus $3 shipping
MC, VISA, AMEX, COD, PO, CHECK

CIRCLE NO. 263 ON READER SERVICE CARD


```
factorial(0,1).
factorial(N,FactN) :- N > 0,
                      M is N-1,
                      factorial(M,FactM),
                      FactN is N*FactM.
```

This is straightforward; the procedure **factorial** calls itself to compute the factorial of the next smaller integer, then uses the result to compute the factorial of the integer in question.

Now consider an iterative algorithm to perform the same task:

```
function factorial(N:integer):integer;
var I,J:integer;
begin
  I:=0;
  J:=1;
  while I<N do
    begin
      I:=I+1;
      J:=J*I
    end;
  factorial:=J
end;
```

In Pascal, this procedure does not call itself. Its Prolog counterpart, however, is a procedure that calls itself as its very last step—a procedure that is said to be *tail recursive*:

```
factorial(N,FactN) :- fact_iter(N,FactN,0,1).
fact_iter(N,FactN,N,FactN).
fact_iter(N,FactN,I,J) :-
            I<N,
            NewI is I+1,
            NewJ is J*NewI,
            fact_iter(N,FactN,NewI,NewJ).
```

Here the third and fourth parameters of **fact_iter** are *state variables* that pass values from one iteration to the next. State variables in Prolog correspond to variables that change their values repeatedly in Pascal.

The recursive clause of **fact_iter** checks that I is still less than N, computes new values for I and J, and finally calls itself with the new parameters. The recursive call is the very last step in this clause; this whole clause is placed last so that when it calls itself, no untried alternatives will be left to save on the stack. This ensures that the stack will not grow during the iteration.

In Prolog (as in arithmetic, but not as in most programming languages), the statement **X is X+1** is always false. Because Prolog variables cannot change their values, the additional variables **NewI** and **NewJ** have to be introduced. **NewI** and **NewJ** contain the values that will replace I and J in the next iteration.

The first clause of **fact_iter** serves to end the iteration when the state vari-

ables reach their final values. A more Pascal-like, but less efficient, way of writing this clause would be:

```
fact_iter(N,FactN,I,J) :- I = N, FactN = J.
```

That is, if I is equal to N, then FactN (which has been uninstantiated until now) should be given the value of J. By writing this same clause more concisely in the earlier procedure, Prolog's unification mechanism is made to perform work that would require explicit computational steps in other programming languages.

Most iterative algorithms can be expressed in Prolog by following the pattern below:

- First, other types of loops (for example, **for** and **repeat-until**) are transformed into Pascal-like **while** loops.
- Then the computation is broken into three stages: the initialization, the loop itself, and any final computations needed to return a result.
- The loop as a tail-recursive clause (for example, the second clause of **fact_iter**) is expressed with the **while** condition at the beginning.
- The final computations is placed in another nonrecursive clause of the same procedure, which is set up so that the nonrecursive clause executes only after the loop is finished.
- Finally, the entire algorithm is hidden behind a "front-end" procedure (**factorial** in the above example), which is what the rest of the program actually calls.

The front-end procedure not only passes along its parameters into the tail-recursive procedure, but the initial values of the state variables as well. The fine art of expressing iteration through tail recursion in Scheme (a LISP dialect) is discussed extensively in *Structure and Interpretation of Computer Programs*, by Harold Abelson and Gerald Jay Sussman (MIT Press, 1985).

Whenever one Prolog procedure calls another, one or more pointers are saved on a pushdown stack. These pointers indicate what remains to be done after return (the *continuation* of the calling procedure) and what alternative solutions remain to be tried (the *alternative set*).

Because every single procedure call places information onto the stack, recursion would appear to lead inevitably to stack overflow. However, several Prolog implementations, including Arity and Turbo, recognize a special case: if both the continuation and the alternative list are empty, nothing need be placed on the stack at all. In this circumstance, instead of calling the next procedure in the normal way, the computer can simply jump into the next procedure without saving a record of where to return. If the procedure is calling itself, this effectively transforms recursion into iteration.

As was stated earlier, a procedure that calls itself with an empty continuation and empty alternative set is described as *tail recursive*; the process of executing such a call without adding any items to the stack is called *tail-recursion optimization* or *tail-recursion elimination*. (The "elimination" of tail recursion does not mean that it should be banished from the program entirely. On the contrary, tail recursion should be used liberally because the implementation transforms it into an efficient, iterative process.)

A quick way to verify that a particular Prolog implementation performs tail-recursion optimization is to try the following predicate:

```
test(N) :- write(N), nl, M is N+1, test(M).
```

Start with the goal **test(1)** and see how long execution continues. If the program runs for more than 10,000 iterations, it is a safe bet that tail-recursion elimination is taking place.

CONTROLLING STACK GROWTH

Recognizing a recursive call that has an empty continuation is easy: the recursive call is the very last subgoal in the clause that contains it. Determining whether the alternative set is also empty takes more thought.

One way to get an empty alternative set is to put the recursive call in the last clause of a predicate. By doing this, the recursive call takes place only after the other alternatives have been exhausted. An example is the longer iterative factorial program that was given earlier. The recursive call takes place only when all other alternatives have been exhausted.

In the third clause of the iterative factorial program, **fact_iter** calls only deterministic predicates (< and **is**) before calling itself recursively. In fact, if this clause had contained any calls to nondeterministic predicates, the alternative set would not have been empty at the time of the recursive call because this same recursive call possibly could have been reached by a different, as yet untried, path.

The alternative set can also be made empty by using the cut operation to rule out alternatives, as in the following replacement for **fact_iter**:

HELP is at hand

USE WITH MS WINDOWS

HELP/Control™ - an on-line help subsystem for the IBM-PC. Increases the value of your software. Save development time and money.

HELP/Runtime. A few simple subroutine calls add context sensitive on-line help to your application. **HELP/Runtime** includes tested interfaces for Microsoft C, Lattice C, Turbo Pascal, IBM BASIC (Interpreter and Compiler), Microsoft FORTRAN, IBM COBOL and assembler. It is distributed with demonstration programs in each language.

HELP/Popup. Add a powerful help system to existing applications, even in dBase or 123, without reprogramming, even without a programmer. It may be memory resident, or, installed with an application, it terminates when the application exits, releasing its memory.

HELP/Generation. Use your favorite editor and our concise screen definition language to build your help files. Compile them into a help system usable by either **HELP/Runtime** or **HELP/Popup.** The package includes sources for sample help files illustrating such features as full-sized or windowed screens.

HELP/Convenience. The screens include highlighted captions. The user selects a caption with the cursor control keys and advances to a new screen, just as with 123.

HELP/Documentation. A detailed manual, both on-line and printed, for the documentation writer and programmer includes instructions which may be incorporated into the user manual.

HELP/Environment. PC-DOS 2.0 or greater is required. **HELP/Runtime** requires approximately 9K for code and buffers for full size help screens.

HELP/Pricing. The complete package (software, both manuals, and demo programs) costs $125.00 and includes a royalty-free license to add **HELP/Runtime** to your applications and to make 25 copies of **HELP/Popup.** A demonstration diskette, including the on-line manual, costs $15.00. A free update to Release 1.1 is available to registered owners. To order, or for more information (including dealer, multiple-copy and site-license pricing) call MDS at 207/772-5436. We accept MasterCard and VISA.

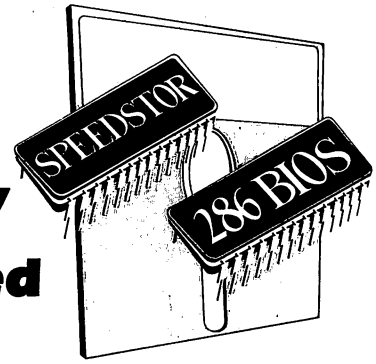
MDS

MDS, INC., P.O. BOX 1237, PORTLAND, MAINE 04104

CIRCLE NO. 146 ON READER SERVICE CARD

Storage Dimension's
SpeedStor 286 BIOS
from Award Software

for use with PC ATs and
XENIX System V
Novell Advanced Netware



- Overcome drive table limitations
- Includes low-level initialization and disk drive advanced diagnostics
- 30% faster than IBM's BIOS
- $129– Retail. Dealer and distributor pricing available

Supports **Seagate ST4096, Miniscribe 6085** and the following:

Seagate	ST251, ST4051, ST4096
Maxtor	XT1085, XT1105, XT1140, XT2085, XT2140, XT2190
Miniscribe	6074, 6085
Micropolis	1325
Control Data	Wren II 94155-86
Microscience	HH1050
Newbury	NDR1085, NDR1140, NDR2190,
Priam Vertex	V150, V170, V185, 519
Toshiba	MK54F, MK56F

STORAGE DIMENSIONS
The Experts in High Capacity PC Storage

408-395-2688
981 University Ave.
Los Gatos, CA 95030

Storage Dimension's family of high performance products for PC storage.
- SpeedStor subsystems from 42MB to 640MB
- SpeedStor hard disk integration software
- 60MB tape backup (DOS, Xenix, Novell)
- SpeedCache caching software
- Extended drive table ROMs
- SpeedStor 286 BIOS

CIRCLE NO. 176 ON READER SERVICE CARD

Attention **Realia COBOL** Users:

SCREENIO 2.0

Screen Manager for COBOL Professionals.

A strictly COBOL approach to screen management, written by and for COBOL professionals. Screen Design is easy with SCREENIO. You take care of your application. We take care of the screens! Fully compatible with Realia COBOL.

What does it do?

Interactive Screen Design, Full Feature Screen Painter, COBOL Data Field Specifications, Edit Masks, Automatic Error Detection, Data Validation, High Speed Video, Full Compatibility, Window Panels, Hot-Fields, Key Redefinition, Foreign Language Capability, Full Color Support, Cursor Management, PC Speaker Control, Screen Libraries, and Very User Friendly. We've thought of everything!

And Best of all:

No Runtime Facility, No Runtime Fees, Superb Support. Only $400 Plus Shipping.
Ask about our FREE demo pack and SCREENIO+Realia Package Deal!

NORCOM
NORTHERN COMPUTING CONSULTANTS
Post Office Box 020897
Juneau, AK 99802-0897
(907) 780-6464
Telex: 5106014951 (NORCOM)
- Amex MC Visa Check -

CIRCLE NO. 191 ON READER SERVICE CARD


```
fact_iter(N,FactN,I,J) :-
    I < N,
    NewI is I+1,
    NewJ is J*NewI,
    !,
    fact_iter(N,FactN,NewI,NewJ).

    fact_iter(N,FactN,N,FactN).
```

Here the recursive call occurs in the first clause, but the cut guarantees that the second clause need not be considered as an alternative. Moreover, it does not matter whether the predicates that were called before the cut are deterministic or not; the cut rules out any backtracking through them.

This technique should be used with caution: it works in some Prolog implementations, including Turbo, but not all. In Arity Prolog 4.0, it appears to work in compiled programs, but, curiously, not in interpreted ones.

Finally, some Prolog packages empty the alternative set under certain conditions by looking ahead to other clauses. For example, the following program is tail-recursive in interpreted Arity Prolog even though it does not appear to be:

```
test(X) :- write(X), nl,
           NewX is X+1,
           test(NewX).

test(0).
```

When X is nonzero, the interpreter looks at the heads of both clauses before trying either. It sees that the second clause cannot work with nonzero X, so it enters the first clause with the alternative set empty. This mechanism, called *indexing*, is seldom well documented and should be tested thoroughly before it is relied upon.

REFERENCES

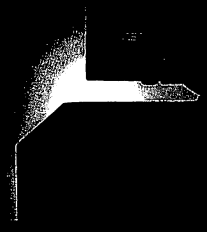
Abelson, Harold, and Gerald Jay Sussman. *Structure and Interpretation of Computer Programs*. Cambridge, Massachusetts: MIT Press, 1985.

Clocksin, W. F., and C. S. Mellish *Programming in Prolog*. Second edition Berlin: Springer-Verlag, 1984.

Kluzniak, Feliks, and Stanislaw Szpakowicz. *Prolog for Programmers*. London: Academic Press, 1985.

Sterling, Leon, and Ehud Shapiro. *The Art of Prolog*. Cambridge, Massachusetts: MIT Press, 1986.

Michael Covington is performing research in automated reasoning and natural language understanding at the University of Georgia.



Announcing AST Premium/286.™

Discover AST Premium/286. The First AST Quality, Uncompromising AT® C

More than two million people have made us the first choice in PC Enhancement. For over five years, you've known AST as the leading PC enhancement company. Now, we're introducing the ultimate enhancement: AST Premium/286. The first AT-compatible personal computer with AST performance and reliability. More flexible and upgradeable. Skillfully combining lightning fast processing speed and uncompromising compatibility.

AST FASTslots:™ Processing speedways. Forming the foundation of the AST Premium/286's increased speed are our FASTslots. This advanced architecture improves overall performance so there's enough built-in power to satisfy even the most demanding user.

The AST Premium/286 operates 50% faster than an 8MHz PC AT® as measured by the Norton Utilities™ Version 3.0 SysInfo. And maintains full compatibility with standard PC and AT-based enhancement cards. It also provides for a powerful, easily upgradeable and expandable future, accommodating the next generation of accelerator and high-performance enhancement cards.

A Heritage Of Software Compatibility. Software compatibility has always been one of our strong points. Shipped with the industry-standard MS-DOS® 3.1, AST Premium/286 is compatible with widely accepted operating systems such as IBM® PC-DOS,™ Concurrent DOS™ and XENIX.™ It's also designed to get the most out of multitasking software packages like Microsoft® Windows, DESQview™ and TopView.™

Applications-oriented. Keyboard-selectable operation at 10, 8 or 6MHz means virtually all popular off-the-shelf IBM PC and PC AT application software is immediately compatible. All your favorites, including Microsoft Word, Lotus® 1-2-3,® Framework,™ Symphony,® dBASE® III and AutoCAD.™

Attain your fullest software potential. AST's advanced architecture also provides faster and more flexible memory addressing. While built-in Enhanced expanded memory capabilities—AST FASTRAM,™ expandable to 2MB in a single slot—let you break the 640K DOS barrier. Create bigger spreadsheets and sort larger databases. And enjoy the uninterrupted workflow benefits of multitasking using current DOS versions, with full support for protected mode software built-in.

Fast access disk storage. Complementing AST Premium/286's speedy operation is a full line of disk systems. There's a 20MB, 40MB and a 70MB hard disk. Both the 40MB and the 70MB offer more storage and faster access times—below 30msec—than the PC AT's fixed disk. And our external disk/tape systems, featuring advanced SCSI architecture, allow easy expandability.

Prices Start At $1995.00*



Personal Computer With Legendary ompatibility and Lightning Speed.

More standards are standard. We build-in our AST FASTRAM™ memory card. And most models include our own multimode enhanced graphics adapter, supporting IBM EGA, CGA and Monochrome, and Hercules Graphics Card™ display modes.

Compatible with AST and IBM Products. AST Premium/286 is designed to remain your productivity partner for years to come. Choose it with confidence for single and multitasking applications, individual and shared environments alike. Use it as an engine with other AST products to form powerful application workstations for desktop publishing, CAD/CAE and more. Or to increase connectivity use it as a network file server, to communicate with IBM mainframes and minicomputers, or to manage multiuser environments.

Solutions that are ready to go. We also offer a number of pre-configured workstation solutions tailored for maximum performance in your application. Combining our proven products, from local area networking and data communications to extra memory and I/O to laser printers and disk systems, our solutions are all designed to increase your business productivity.

Quality across the board, around the world. When you buy AST products, you're also purchasing a worldwide reputation for service, support and product dependability. AST Premium/286 is backed by a one year limited warranty, and our worldwide network of certified dealers and service centers.

AST Premium/286—The system and the solutions. For more information call our Product Information Center at **(714) 863-0181** or send the coupon to: AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714-4992.

Yes, I want to know more about AST Premium/286 Solutions. Send me more information today.

Name:_____

Title:_____

Company:_____

Address:_____

City:_____ State:_____

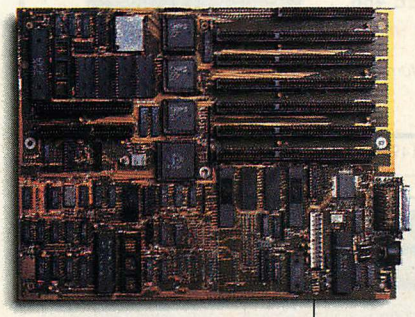
Zip:_____ Telephone: ( )_____

My applications:

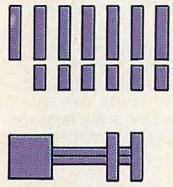
____ Desktop Publishing ____ Multiuser
____ Mainframe/Minicomputer Connection
____ Multitasking ____ General Business

Send to:
AST Research, Inc., 2121 Alton Avenue, Irvine, CA 92714-4992 Attn: M.C. PCTJ3/87
01PCTH000B01PM

AST RESEARCH INC.



Seven industry-standard expansion slots; 1 PC-compatible slot, 6 PC AT-compatible slots, including two AST FASTslots. FASTslots provide no wait state *operation with a high-speed direct interface to the 10MHz 80286 processor. Advanced architecture accommodates the next generation of accelerator and high-performance enhancement cards. It's also an open architecture for easy development and system integration.*

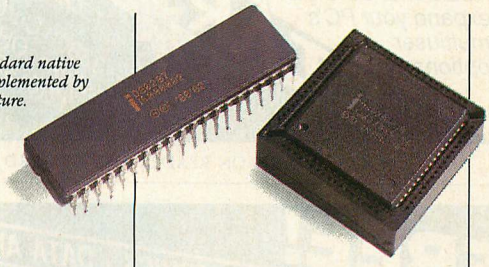


Two AT-compatible expansion slots with a plus: a third bus connector featuring lightning-quick CPU access time, for use with specially-designed cards like the AST FASTRAM Enhanced memory card. Expandable to 2MB in single slot, FASTRAM supports a variety of addressing capabilities—Enhanced EMS, EMS, extended (protected mode) and conventional memory addressing.

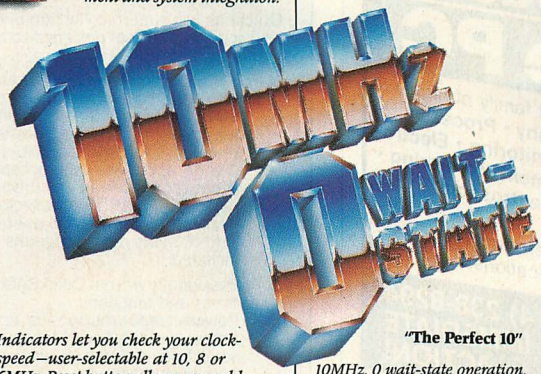
Enhanced, low-profile 101/102-key keyboard with separate numeric keypad, dedicated cursor control and extra function keys. International versions available.

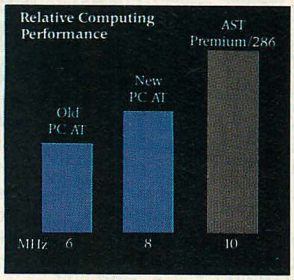
AST Premium/286 is shipped with MS-DOS and GW BASIC,® and it's fully compatible with a wide variety of operating systems, operating environment and utility packages, and application software.

Based on industry-standard native 80286 technology, complemented by AST advanced architecture.



Coprocessor socket accepts 8MHz 80287 devices to execute math- and floating point-intensive programs faster.





Compare the increased speed of the AST Premium/286 against the top competitors for yourself. (Basis: Norton Utilities SysInfo Version 3.0)

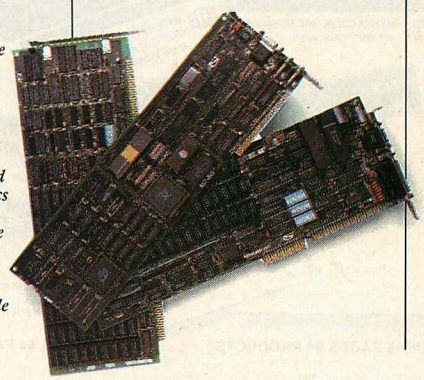
Indicators let you check your clock-speed—user-selectable at 10, 8 or 6MHz. Reset button allows easy cold-booting. Security lock prevents unauthorized keyboard access.



"The Perfect 10"

10MHz, 0 wait-state operation, faster than the 8MHz PC AT, with IBM PC AT hardware and software compatibility.

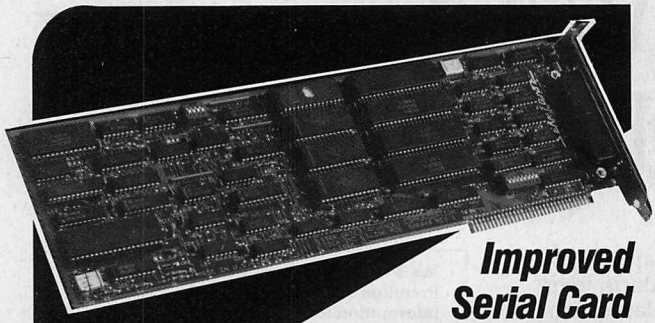
Supports standard PC, PC AT and AST FASTslot cards. With AST, you start with a lot, like our included FASTRAM Enhanced memory card and multimode Enhanced Graphics Adapter, but you've also got a lot of options. We offer the widest range of compatible enhancement products and peripherals to suit your needs—a true one-stop solution allowing you to expand and upgrade your system with the assurance of future service and support.



AST markets products worldwide—in Europe call: 44 1 568 4350; in the Far East call: 852 0499 9113; in Canada call: (416) 826-7514.

AST Premium/286, FASTslot and FASTRAM trademarks of AST Research, Inc. IBM, Personal Computer AT and PC AT registered trademarks and PC-DOS and TopView trademarks of International Business Machines Corporation. Lotus, 1-2-3 and Symphony registered trademarks of Lotus Development Corporation. dBASE registered and Framework trademark of Ashton-Tate. AutoCAD trademark of AUTODESK, Inc. Microsoft, MS-DOS, XENIX, and GW BASIC registered trademarks of Microsoft Corporation. DESQview trademark of Quarterdeck Office Systems. Hercules Graphics Card trademark of Hercules Computer Technology. Norton Utilities trademark of Peter Norton Computing, Inc. UNIX trademark of AT&T Bell Laboratories, Concurrent DOS trademark Digital Research. Copyright © 1986 AST Research, Inc. All rights reserved.

CIRCLE NO. 126 ON READER SERVICE CARD



Improved Serial Card ...EXTENDS SPEED AND FLEXIBILITY OF PC APPLICATIONS

New improved 8-channel ACL™ serial card now offers the option of RS422 and RS485 compatibility.

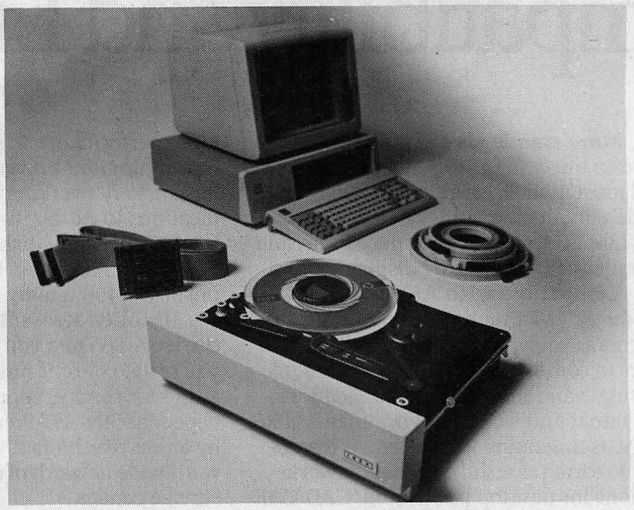
The new card provides all the standard ACL "smart card" features...flexible address decoding, programmable interrupts, expandable options via software, and the ability to download custom programs...now the ability to operate over greater distances and speeds of RS422 applications. In addition, the new card enables PCs to interface with industrial process controls via RS485 links.

Call for free literature that explains how the improved ACL Serial Card can expand your PC's multiuser options.

STAR GATE TECHNOLOGIES INC.
33800 Curtis Blvd.
Eastlake, OH 44094
(216) 951-5922

CIRCLE NO. 177 ON READER SERVICE CARD

9-TRACK MAG. TAPE SUBSYSTEM FOR THE IBM PC/XT/AT AND...



For information interchange, backup and archival storage, IBEX offers a 9-track, IBM format-compatible ½" magnetic tape subsystem for the IBM PC, featuring:

- **IBM format 1600/3200 and 800 cpi.**
- **Software for PC-DOS, MS-DOS.**
- **Also for DEC, VAX, VME, S-100, RS-232, IEEE 488.**

Write, phone or TWX for information.

IBEX

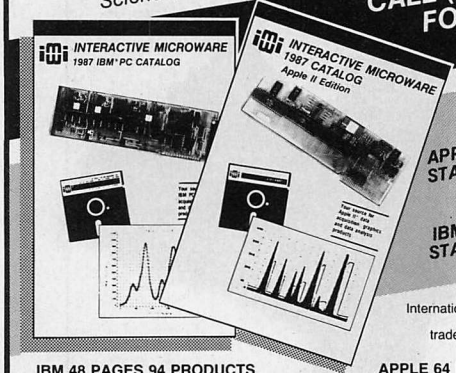
IBEX COMPUTER CORP.
20741 Marilla St.
Chatsworth, CA 91311
(818) 709-8100
TWX: 910-493-2071

CIRCLE NO. 149 ON READER SERVICE CARD

FREE! DATA ACQUISITION • GRAPHICS • DATA ANALYSIS PRODUCT CATALOGS FOR YOUR IBM or Apple PC

Interactive Microware offers a complete, low cost family of high quality hardware and software for **Chromatography • Process Control • Spectrophotometry • Temperature Monitoring • Electrochemical Analysis • Atomic Absorption • Instrument Monitoring • Data Plotting • Curve Fitting • IR and UV/VIS Spectroscopy • Computer Assisted Design** and Much More!
Thousands of clients worldwide use IMI products for Science, Engineering and Business Applications.

CALL (814) 238-8294 FOR IMMEDIATE ACTION!



featuring

COMPLETE APPLE II WORKSTATIONS STARTING AT $2220

COMPLETE IBM PC WORKSTATIONS STARTING AT $3595

IBM is a registered trademark of International Business Machines Corp; Apple is a registered trademark of Apple Computer, Inc.

IBM 48 PAGES 94 PRODUCTS

APPLE 64 PAGES 264 PRODUCTS

imi™

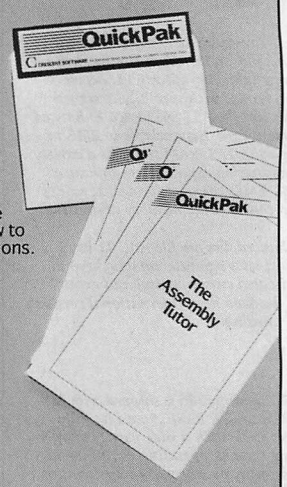
INTERACTIVE MICROWARE, INC.
POB 139, Dept. 237, State College, PA 16804
Phone: (814) 238-8294 · Telex 705250

CIRCLE NO. 208 FOR APPLE, 209 FOR IBM

QuickBASIC just got quicker with QuickPak

QuickPak is a superb collection of enhancements, subroutines, and instructional material designed to help you get the most out of programming in BASIC.

- Powerful assembly language routines to give your programs more speed, more power, and full access to DOS and BIOS services. SORT all or part of a string array with one command! Complete windowing capability — display help screens instantly, overlay text. FIND any string or sub-string within an entire array *regardless of capitalization* — accepts wildcards. READ directories into your programs from any drive or path. READ/WRITE disk sectors — create your own DOS utilities! MANY, many more programs included.
- Professionally written QuickBASIC routines and functions. Powerful input routines for text, dates, and numbers. Menus, scroll bars, date/time functions, and much more.
- The Assembly Tutor — a complete guide to learning assembly language *from a BASIC perspective.* Learn how to create your own routines and extensions.
- Tips and Tricks book — packed with clever ideas and techniques to help *you* be a better programmer.



You get all this, all of the source code for *every program included,* and a thirty-day money back guarantee for only $69.00.

No royalties are required for using any of the QuickPak routines in your programs. Not copy protected, of course.

by

CRESCENT SOFTWARE
64 Fort Point Street, East Norwalk, CT 06855
(203) 846-2500

QuickPak requires Microsoft QuickBASIC or BASCOM, DOS 2.0 or higher. Visa, M/C, C.O.D., or checks accepted.

CIRCLE NO. 145 ON READER SERVICE CARD

Reviews and Updates



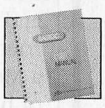
REPERTOIRE
PMI



MACE UTILITIES
Paul Mace Software



FILEMAX
Grand Max Software Corporation

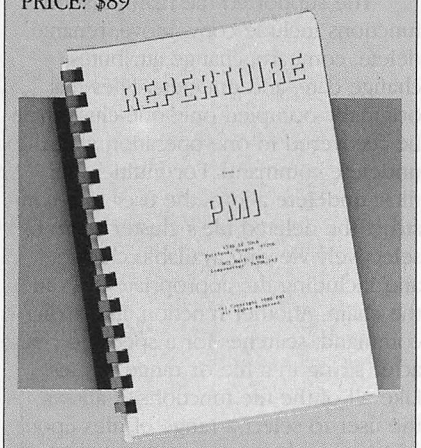


BASTOC
JMI Software Consultants, Inc.

REPERTOIRE, version 1.3

PMI
4536 S.E. 50th
Portland, OR 97206
503/777-8844

PRICE: $89



CIRCLE 351 ON READER SERVICE CARD

Modula-2 systems have not engendered the same toolbox generation that pervades the C and Turbo Pascal markets. Typically, most Modula-2 compilers come with a set of skeletal libraries. Users then create their own sets of higher level routines for screen management, extended I/O services, and graphics. To fill part of that need, PMI has introduced REPERTOIRE, a collection of high level tools for Modula-2 programmers. This package contains a screen management package with natural language analysis facilities, a screen editor, and a window oriented editor. Indexed file I/O and low-level services complement the package.

REPERTOIRE is designed to support two Modula-2 compilers—Interface Technologies Corporation's M2SDS and Logitech, Inc.'s Modula-2. (For a review of these products, see "Modular Development," John T. Cockerham, this issue, p. 114.) PMI delivers the routines on two diskettes or as a set of four for both compilers for an additional $20. The installation routines are well documented in the manual.

The REPERTOIRE manual is very complete and includes a comprehensive index. The writing is exceptionally clear. The copy, however, is less clear. The type is blurred and microscopic.

In a radical departure from most software companies, PMI also includes the Modula-2 source code in the base price. The author's stated purpose is to allow the user to support some of the code and reserve the difficult bugs for company support.

The use of the screen system starts by creating the screen template with a basic text editor. This template is a series of lines that indicate to the screen manager the screen number, data fields, colors, acceptable user responses, and the next screen in sequence. The specifications are fed to the screen compiler (its source code is included with the package) that creates a .DSP file. The programmer calls up a screen by calling the screen manager and specifying the .DSP file and the screen number. The screen manager interprets the contents of the .DSP file. Included with the screen manager is a simple, screen-oriented text editor that allows a uniform user interface for text entry.

The natural language facilities are actually a pattern expression evaluator. The pattern can include Boolean operations on the presence or absence of a string from the test string, the length of the test string, and the relative positions of words within the test string. The routine returns an index into the pattern string where the match occurred. For example, pattern string

'chicken' & 'pot' \ 'garage' & 'car'

defines two patterns. The first searches for *chicken* and *pot* in a string; the second searches *garage* and *car*. The expression evaluator returns 0 for no match, 1 for a match of the first pattern, and 2 for the second. For example:
A chicken in every pot returns 1.
Two cars in every garage returns 2.
Chicken potato pie returns 1.
A chicken in every car returns 0.

An option is provided to match only whole words, and the word-delimiting characters may be specified by the user. With appropriate user-written patterns, the string expression evaluator can be used to create a simple natural-language command interface.

PMI has written these routines to accommodate a multitasking environment. The windowing and screen-oriented text editor can be invoked by several user-written Modula-2 processes simultaneously; several editing windows can be open. Text exchange can be performed between the windows.

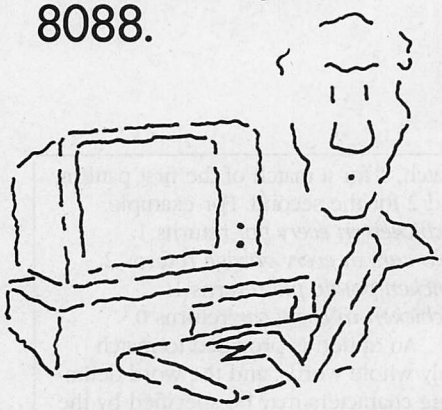
With version 1.3 of REPERTOIRE, PMI introduced an indexed file system. This system allows variable-length records and is quite fast. The file manager keeps the linked-list file index in RAM at all times, searching through RAM for the desired record. The index is updated only when the file is closed. An optional programmed mode saves the index with every update to the file. If the index is totally lost, a recovery mechanism automatically runs when the file is reopened, regenerating the index.

To overcome the differences between the two Modula-2 systems' implementations of low-level language facilities, PMI includes its own low-level routines. These routines fill in where the two implementations fall short. For the Logitech system, the routines provide long cardinal and long integer routines. For the Interface Technologies' system, the routines provide alternative implementations of the string functions.

The screen system is for text only. The text editor uses the F10 key to signal the end of data entry rather than the Enter key. Using the source code, this is a trivial change to make.

Get a Grip on Assembly Language.

The award winning Visible Computer: 8088.



The Visible Computer is a book and software combination for mastering the elusive skills of assembly language. PC Tech Journal took one look and made it their September '85 "Program of the Month."

It's an animated simulation of the PC's microprocessor that lets you see with your own eyes how assembly language works. You'll be using it as a debugging tool for years to come.

It's a tutorial. A lot of people think the 350 page manual is the best book on assembly language ever written.

It's 45 demonstration programs you'll execute with the simulator, from simple register loads to advanced programs that manipulate interrupts and perform file I/O. And what you'll learn applies to all 86 family processors, including the 80186 and 80286.

$79.95
not copy protected

The Visible Computer for IBM PC/XT/AT and true compatibles. If your dealer doesn't have it, order direct: Software Masters, 2714 Finfeather, Bryan, TX, 77801. (409) 822-9490. Please include $3.00 shipping. Bank cards accepted.



TVC takes you inside the processor as it executes programs.

Software Masters™

CIRCLE NO. 168 ON READER SERVICE CARD

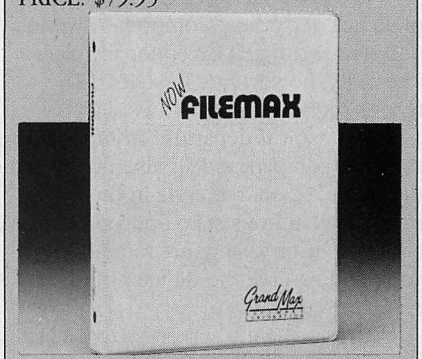
In today's marketplace, very few manufacturers will take the risk of supplying the source code with their products, especially one as well conceived and well executed as REPERTOIRE. The routines offered in this package provide a Modula-2 developer with a useful set of high-level tools that are not available in the base language. They represent an outstanding bargain.

—JOHN T. COCKERHAM

FILEMAX

Grand Max Software Corporation
P.O. Box 10580
Portland, OR 97210
503/226-0808

PRICE: $79.95



CIRCLE 352 ON READER SERVICE CARD

With the proliferation of hard disks comes an increased interest in programs that organize and manage hard-disk files and directories. FILEMAX is a disk management utility with an impressive set of features, including directory tree copying, directory renaming, file recovery, and disk optimization.

Unlike many popular utilities, FILEMAX is not a terminate-and-stay-resident program; it is intended to be invoked periodically, as needed, for directory viewing, organization, and clean up. An installation utility is provided to allow the user to modify program characteristics such as keyboard set-up, sound (on/off), and help file path. The FILEMAX program is not copy protected.

FILEMAX uses full-screen output and function-key command driven input. The output screen is split into two areas: the top 19 lines display outputs and the bottom 6 lines display prompts. A small menu in the lower left screen corner lists valid function keys and describes their use. On-line help is always accessible in FILEMAX with the F1 key. The help screens provide brief descriptions of options that are available at any point in the menu hierarchy.

FILEMAX's functions are in three categories: directory manipulation, file manipulation and display, and manipulation of the file allocation table (FAT).

Among the directory functions are commands to move, rename, create, delete, and sort directories. The move and delete functions are particularly powerful; using them the user can modify entire directory chains. The **move directory** command allows a directory to be inserted before, after, or within, that is, as a child subdirectory to any other directory. The **delete directory** command requests verification that the directory's files and all of the subdirectories are to be deleted. With the **sort** command directory entries can be sorted on any combination of name, extension, date, time, size, or attributes in either ascending or descending order by the sort field.

The supported file manipulation functions include copy, move, rename, delete, compare, change attributes, change date, and undelete. Files that originally occupied only one cluster can be recovered in one operation with the **undelete** command. For multicluster files, **undelete** allows the user to reconstruct the deleted file's cluster chain by selectively viewing available clusters and including the appropriate ones in the chain. Another function, **find string** command, searches for a specified character string in a file or range of files. Like all of the file functions, it allows the user to select a range of files upon which to operate.

The FAT analysis portion of FILEMAX provides a well-designed display of the FAT that shows the chaining of clusters in files, open clusters, bad clusters, and cross-linked clusters. The contents of any cluster can be viewed in either text or hexadecimal format.

An option called "tune up" allows the clusters on the disk to be reorganized in such a manner that all files are contiguously allocated on the disk and all subdirectories are placed at the beginning of the disk. However, the tune-up feature of the tested version reflects a possibly serious design flaw. As clusters are rearranged on the disk, FILEMAX keeps track in RAM of the corresponding changes to the FAT chaining and directory entries. This information is written to disk only at the completion of the tune-up operation. Any unexpected interruption of the operation, for example, from power loss or accidental rebooting can result in significant damage to the file structure of the disk, perhaps rendering it unreadable.

FILEMAX 3.42 contains another annoying bug that shows up when drive B: is referenced on a single-diskette-drive system. The program produces an error box that contains the message GX I/O ERROR and a display of registers, at which point the machine has to be rebooted. Using the **assign** command to assign B: to A: causes a similar problem.

FILEMAX has difficulty coexisting with some pop-up menu programs and device drivers. When Hersey Micro Consulting's FANSI-CONSOLE or the WordPerfect Library Calendar was installed before FILEMAX was started, the system hung when FILEMAX was exited. When FILEMAX is running, it directly uses the keyboard interrupt and does not pass through unrecognized keys (by calling the previous keyboard vector address).

The documentation for FILEMAX consists solely of a 19-page overview of program operation. Considering the complexity of the program and its capabilities, this is rather sparse information. Users familiar with DOS disk and directory structure, and who have used other disk utilities, such as the Norton Utilities, will make effective use of FILEMAX. However, the hierarchy of menus and functions is complex; even the veteran programmer will need time to master program operation.

—GLENN ROBERTS

MACE UTILITIES, version 4.0

Paul Mace Software
123 N. First Street
Ashland, OR 97529
800/523-0258

PRICE: $99



CIRCLE 353 ON READER SERVICE CARD

Mace Utilities from Paul Mace Software will be of special interest to anyone who has ever accidentally formatted a hard disk that contained valid data. Mace Utilities can reconstruct a hard disk after a format, and more.

The IBM description of the DOS FORMAT command has a dramatic caution warning users that formatting destroys all of the data on the target disk. However, the *DOS Technical Reference* manual states, "When formatting a fixed disk, FORMAT checks all locations within the DOS partition, but does not physically format them again."

The FORMAT procedure consists of two distinct operations: physical formatting and logical formatting. Physical (or low-level) formatting, which is performed only on diskettes, builds the skeleton (the so-called *sectors*) onto which a logical DOS disk structure is built. The hard disks are already physically formatted when shipped by IBM. Logical (or high-level) formatting of a hard disk actually consists only of analyzing it for any defective areas, and of preparing it to accept DOS files by initializing the root directory and the file allocation table (FAT).

IBM's caution is not entirely accurate because FORMAT does not destroy

TAKE COMMAND.

INTRODUCING THE SURE-FIRE WAY TO TAKE COMMAND OF MS-DOS
INTRODUCING COMMAND PLUS ONLY $79.95

Now you can take command of MS-DOS with Command Plus, the programming shell and command processor that replaces COMMAND.COM in Versions 2.0 and above.

What this means is that you can now program efficiently in DOS without having to learn UNIX.

Programming efficiently means that Command Plus will increase your programming productivity and flexibility by virtually eliminating the repetitive tasks DOS COMMAND requires.

For example, Command Plus' aliasing function lets you create fast, memory resident macros, while the history command lets you use the cursor keys to recall, edit and execute commands that you ran up to 48 command lines ago.

What's more, Command Plus also equips you with Script, a batch processor that's easy to learn and unbelievably powerful. It's Pascal-like language includes control loops, conditionals and variables which let you create unique systems utilities.

In short, Command Plus means that you can now program easily and efficiently in the familiar DOS without having to learn UNIX.

It's no wonder that Ray Duncan has already called Command Plus "a very well designed and sound product."

And has written that "The regular expression support, the enhanced copy, move and del commands, and the browse utility are particularly neat."

At only $79.95, plus shipping and handling, the price of Command Plus is pretty neat, too.

So, if you want a sure-fire way to take command of DOS, or if you want more information about Command Plus, call us at (800) 992-4ESP. In California, call (213) 390-7408. VISA and MasterCard accepted.

MS-DOS is a registered trademark of Microsoft Corporation.

(800)992-4ESP (213)390-7408
COMMAND PLUS.
ESP
SOFTWARE SYSTEMS INC.
11965 Venice Boulevard
Suite 309
Los Angeles, CA 90066
ESP

CIRCLE NO. 190 ON READER SERVICE CARD

a hard disk's data area. The root directory and FATs are cleared with zeros, but data previously placed on the disk are still there and may be recovered.

Because the content of a subdirectory is stored as a file within the data area, subdirectories are not destroyed by FORMAT and also can be recovered. All that is lost during a format is the original root directory pointer to the subdirectory "file." A well-designed program can examine a freshly formatted disk and find "files" that are probably subdirectories. The Mace Utilities package is designed to do just that.

Mace Utilities consist of one program (RXBAK.EXE) that is to be executed periodically and another series of programs that are driven from a menu system. RXBAK takes a snapshot of the hard disk's DOS partition boot sector, the FATs, and the root directory. This snapshot is copied into a special file that is stored at the logical end of the DOS partition. If a hard disk is accidentally formatted, Mace can copy this file back into the respective sectors from which it was created, thus restoring the directory to the state it was in the last time RXBAK was run. In most cases, this will allow the reclamation of some, if not all, of the lost data.

Because RXBAK takes only a few seconds to run, a call to it can be placed within any frequently used batch file. It can preserve many hours of work by safeguarding critical data structures. Note that these file recovery methods are only salvage operations, and because of that, the restored disk should be formatted once again after the recovered files have been safely transferred to another device.

Even if it is not installed on the hard disk, Mace can be run from the diskette to recover data lost due to an accidental FORMAT. Mace is not able to recover any files that were in the root directory. They probably still do exist; however, all pointers have been lost as a result of the clearing of the directory and FAT. Although Mace does not know where DOS placed subdirectory files, it does know in general what they look like. Mace searches the entire disk looking for possible subdirectory candidates. Once subdirectories are located, all files and subdirectories can be partially, if not fully, recovered. Restored first-level subdirectories will appear in the reconstructed root directory as SUB1, SUB2, SUB3, etc. Inside each SUB*n*, lost files can be found.

Mace Utilities contain several other useful functions for both diskettes and hard disks; these include the usual **file undelete** function and the more ambitious **diagnose**, **remedy**, **squeeze/sort directories**, and **condense** functions. **Diagnose** attempts to read every sector of a disk looking for errors. If it finds a sector that it cannot read correctly and that has not been previously marked as bad (in the FAT), it issues a message advising that this file should be attended to at once. **Diagnose** does not repair the disk; the **remedy** function is designed for that purpose.

Remedy is similar to **diagnose**, except that unreadable files are made readable with a minimal loss of data. If an unreadable sector is found within a cluster that is not allocated, it is locked out (in the FAT). If, however, an unreadable sector is part of some file, all readable sectors within a cluster are reconstructed and copied to another free cluster. The unreadable sector is filled with asterisks (*), and the FAT chain is relinked. Finally, the cluster containing the bad sector is locked out. If the file

The Advanced Programmer's Editor
That Doesn't Waste Your Time

EPSILON

- Fast, EMACS-style commands—completely reconfigurable
- Run other programs without stopping Epsilon—concurrently!
- C Language support—fix errors while your compiler runs
- Powerful extension language
- Great on-line help system
- Multiple windows, files
- Regular Expression search
- Unlimited file size, line length
- Supports large displays
- 30 day money-back guarantee
- Not copy protected

Only $195

Lugaru Software Ltd.

5740 Darlington Road
Pittsburgh, PA 15217

Call
(412) 421-5911

for IBM PC/XT/AT's or compatibles

CIRCLE NO. 125 ON READER SERVICE CARD

that contains the bad sector is a .BAT, a .COM, or a .EXE file, an error message is issued to the effect that this program probably will no longer execute properly and should be replaced. It should be noted that the actions taken by **remedy** are permanent; therefore, it normally is run only after **diagnose** has been used to gather information about the damaged files.

Squeeze/sort directories sorts the entries within a directory on one of the four fields (name, extension, date/time, or size) and moves all active directory entries to the top of the directory. This allows DOS to search through directories without the time-consuming process of reading and analyzing entries for nonactive (deleted) files.

As an option, **squeeze/sort** can set a flag within directory entries of .BAT, .COM, and .EXE files to "read only." This prevents them from being deleted (this effect can be reversed with the DOS command ATTRIB) and also enhances the performance of subsequent runs of the **condense** option.

Condense performs all of the diagnostic and repair operations described above, and then rearranges the files into a linear space. When it is finished, the files are contiguous (with the exception of bad sectors) and packed from the front of the disk. The order of packing is directories, read-only files, and then all other files. Directories are now close to the root, which means the seek time will be minimized. Most executable files are never operated on; they are merely read. **Condense** then packs all of the read-only files to the front of the disk, where they are close to the subdirectories and in one contiguous segment. This implies that they can be accessed rapidly.

The utilities Vcache, Vkette, and Vscreen have been added to version 4.0 of Mace. These programs are designed to speed up hard disk and diskette transfers and output to video displays.

The documentation with Mace Utilities is adequate, and in some cases it provides more information than users will want. On-line help is available; estimated runtimes as well as function descriptions are provided.

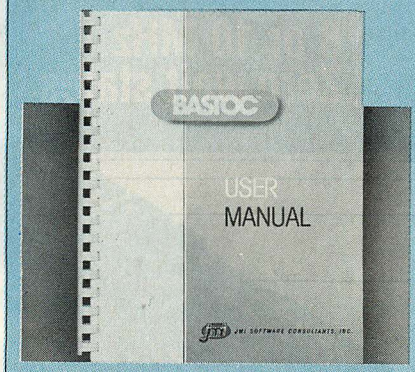
For $99, users can invest in a fairly comprehensive insurance policy for recovering lost files. Mace Utilities may not be able to recover all of the information, but the product is able to salvage a great deal of lost data, and that is a worthwhile dividend.

—GUY QUEDENS

BASTOC

JMI Software Consultants, Inc.
904 Sheble Lane
P.O. Box 481
Spring House, PA 19477
215/628-0840

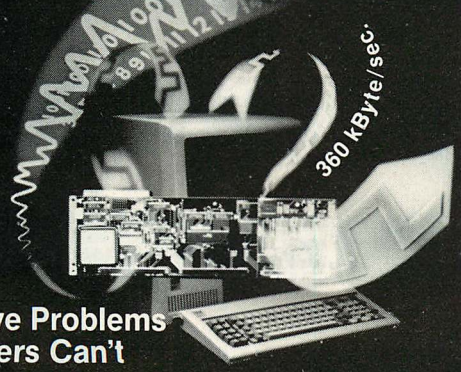
PRICE: $495



CIRCLE 354 ON READER SERVICE CARD

As production-quality C compilers have become available in the last few years, many programmers saddled with maintaining large BASIC systems have developed a keen interest in migrating software as painlessly as possible to C, while rewriting a minimum of code. This conversion need is

Only Now... DMA Analog, Digital and Counter Data... FAST!



Solve Problems Others Can't

Now, your PC-based data acquisition and control systems can capture, analyze, and react to real-world events... in real time. Our new DMA plug-in carrier boards can help you create powerful, IBM-compatible PC systems... economically.

- Modular magic. Configure a few points, or sophisticated systems up to hundreds of channels.
- Pre- and post-event capture and foreground/background.
- High-speed software in BASIC, C, Turbo Pascal, and ASYST.

Call or write for complete information and applications assistance. Burr-Brown Corp., P.O. Box 11400, Tucson, AZ 85734. 602-746-1111.

High Performance PC Data Acquisition Systems

BURR-BROWN®

CIRCLE NO. 117 ON READER SERVICE CARD

PMi FASTCARD®
INTRODUCES with 2 MBYTES



ONLY $295*

The Only Expanded Memory Card Which Can Save A Day Each Week.

A major independent testing laboratory benchmarked a PC-XT equipped with *FASTCARD* and demonstrated a tenfold improvement over a PC-AT on typical Disk I/O operations.

- Portable between IBM PC, XT, AT and compatibles.
- Up to 2MB with Split Memory Mapping to
  - Fill memory to 640K
  - Provide Expanded memory over 640K
- Unique Disk Caching
- Ram Disks (up to 8MB)
- Custom Password Security
- Print Buffering
- Built-in Diagnosis and Automatic Fault Tolerance
- Factory Installed and Tested DRAM's
- Lotus/Intel/Microsoft compatible

*Each *FASTCARD III* comes with 2 MBytes of Memory. *FASTCARD IV*, available with 2 MBytes, includes serial/parallel ports, game port and a clock calendar for $295. Dealer inquiries invited.

LOTUS EMS COMPATIBLE

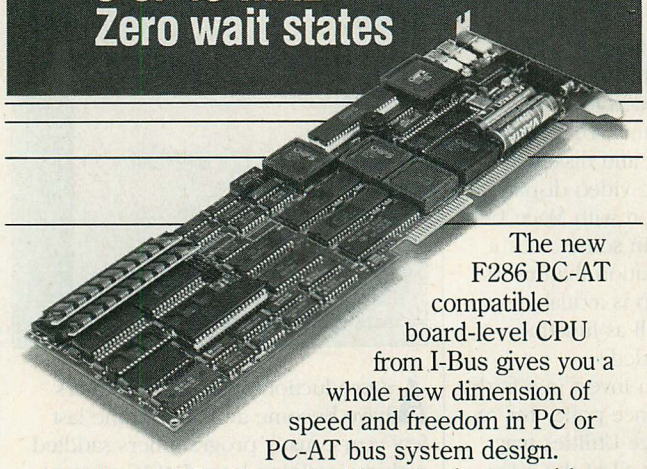
For additional information, contact PMi at

PERIPHERAL MARKETING INC. **602-483-7983**
7825 E. EVANS RD., #500, SCOTTSDALE, AZ 85260

CIRCLE NO. 202 ON READER SERVICE CARD

PC-AT BUS BOARD-LEVEL COMPATIBLE

8 or 10 MHz Zero wait states



The new F286 PC-AT compatible board-level CPU from I-Bus gives you a whole new dimension of speed and freedom in PC or PC-AT bus system design.

It's all on a PC add-on-sized board—for use with a passive backplane just like other board-level systems. You just add the expansion cards, put it in a box (I-Bus has loads of backplanes and boxes), and it's ready to execute any PC-AT applications software.

Use the F286 in a disk-based or diskless system, with or without a keyboard, with or without a display.

It's packed with features such as 10 MHz zero wait state operation. Separately clocked 80287 support (runs at full speed—not half speed as in other AT's). 512K RAM. Battery-backed clock/calendar. Optional PROMDISK to run any application from the F286's user EPROM.

And best of all, it's designed, built and supported by I-Bus—the originators of the passive backplane PC Bus.

If you're into systems, we speak your language. Call us TOLL FREE at:

800-382-4229

(in CA call (619)569-0646)



I-BUS

The Full Service PC Bus Company

5780 Chesapeake Court
San Diego, CA 92123 TLX: 910 240 0290

CIRCLE NO. 210 ON READER SERVICE CARD

FIGURE 1: *BASIC Sieve of Eratosthenes*

```
090 REM       for PC-BASIC, BETTER, PROFESSIONAL
095 REM
100 DEFINT A-Z
110 LIM = 8190
120 DIM FLAGS(8191)
130 PRINT TIME$
140 FOR I=0 TO LIM
150   LET FLAGS(I) = -1
160 NEXT I
170 PRINT TIME$
180 FOR I=0 TO LIM
190   IF FLAGS(I) = 0 THEN 270
200   PRIME = I+I+3
210   LET K = I+PRIME
220   WHILE K <= LIM
230     LET FLAGS(K) = 0
240     LET K = K + PRIME
250   WEND
260   LET COUNT = COUNT + 1
270 NEXT I
280 PRINT COUNT; "PRIMES"
290 PRINT TIME$
```

The classic Sieve of Eratosthenes benchmark program was used as a simple test of BASTOC's conversion capabilities.

FIGURE 2: *BASTOC-Translated Sieve in C*

```
char *TIME_();
static int FLAGS[8192], count, i, k, lim, prime;
static int it_1, it_2;
static char *st_1;
main(argc, argv)
    int argc;
    char *argv[];
    {
    bio_init(argc, argv, 1);
    /* for PC-BASIC, BETTER, PROFESSIONAL */

    lim = 8190;
    BPRINT("s", TIME_(&st_1));
    it_1 = lim;

    for (i = 0; i <= it_1; ++i)
     {
     FLAGS[i] = -1;
     }
    BPRINT("s", TIME_(&st_1));
    it_2 = lim;

    for (i = 0; i <= it_2; ++i)
     {
      if (FLAGS[i] == 0)
         goto l_270;
     prime = i + i + 3;
     k = i + prime;
     while (k <= lim)
         {
         FLAGS[k] = 0;
         k = k + prime;
         }
     count = count + 1;

l_270:;
     }
    BPRINT("i;s", count, "\006PRIMES");
    BPRINT("s", TIME_(&st_1));
    bexit(0);
    }
```

The IF-THEN of the BASIC program was changed to a C goto statement. The rest of the program is well structured.

addressed by BASTOC, a software package from JMI Software Consultants, Inc. BASTOC accepts a BASIC program in source form as input and creates equivalent C source code from it. (BASTOC was originally reviewed in "BASIC to C," Ernest Tello, October 1984, p. 117.)

BASTOC is intended as both a migration utility (making a one-time conversion of a program from BASIC to C) and as a production BASIC compiler that produces a C program as its output (for use in developing prototype or throwaway programs). The BASTOC package consists of a translator program and a runtime library in small and large memory models. The library contains support for dynamic string handling (a feature not normally found in C), screen control, and BASIC-style file I/O. The module ANSI.SYS must be present to use BASTOC's screen routines.

C source code is provided for the runtime library, several translator functions, and data tables. The package is shipped with four diskettes and an 8½-by-11-inch spiral-bound manual. BASTOC is not copy protected.

Installation of the program is a straightforward procedure that is well documented in the manual. It consists of copying the files on the supplied diskettes to a single directory. Following installation, the directory includes two versions of the translator program, one that simply produces a C source file as output and another that acts as a complete compiler, calling the Microsoft C compiler and linker directly after source translation. Also included in the directory are the runtime library in the appropriate memory model and a main program that handles intra-program chaining and COMMON variables.

BASTOC supports the BASIC.EXE (not the advanced BASICA) version of Microsoft BASIC for the PC. Statements that are not supported include PEEK, POKE, and VARPTR for large memory models (implying that assembler language subroutines need manual conversion), and all of the BASICA graphics and sound statements. The CHAIN, COMMON, RUN, and SHELL statements also have some restrictions. For most business software, this should prove to be an adequate subset of BASIC.

Notably absent from BASTOC is support of the more recent Microsoft QuickBASIC and IBM BASIC Compiler version 2.0 enhancements, such as separate subroutines and ISAM files. In its current version (2.1c), however, BASTOC has added support for some features it previously rejected: NEXT instead of NEXT *variable*, use of % for specifying integer variables, and READ...DATA.

The BASTOC options control code optimization (including detecting and deleting unreachable code), default variable types, program organization, and subroutine handling (GOSUBs can be left in-line in a single, monolithic function or isolated into separate C functions). An new option in version 2.1c optimizes expressions by using C's increment and decrement operators wherever possible. Wild card file processing is not available.

The code translated by BASTOC for the BASIC version of the Sieve of Eratosthenes (see "The State of C Interpreters," Marty Franz, May 1986, p. 153 and "Reconsidering BASIC," Marty Franz, December 1986, p. 142) is shown in figure 2, with the original BASIC version in figure 1. A C **goto** statement and matching label have been added where the GOTO existed in the original program. The TIME_ function may need to be rewritten for some compilers. In general, the resulting C program is well-structured and fairly portable.

Within the above framework, operation of BASTOC is straightforward and produces no surprises when the Microsoft C compiler and linker are used. Other compilers may require modification to the runtime library. Most of the library functions are available in source form and are adequately documented by JMI; a knowledge of C is required to make the modifications. The more unstructured the BASIC source code is, the more unstructured the C output will be, especially when GOTOs and GOSUBs are used carelessly. A disciplined BASIC coding style is the way to ensure that the C code generated by BASTOC can be maintained.

An important feature of the BASTOC package is the translator, which can be modified and rebuilt to handle additional functions and statement types. This gives it the potential for translating other BASICs, such as QuickBASIC, into C. Hooks have been provided into the translator's key-word tables and recognition routines to allow user-written functions to be added, but this has not been well-documented in the manual and must be considered a technical challenge to even an experienced C programmer. Nonetheless, it should prove easier to extend BASTOC than to write a comparable translator from scratch. BASTOC is a competent programming tool for programmers converting large BASIC systems to C.

—MARTY FRANZ

CQL

SQL Compatible Query System adaptable to any operating environment.

CQL Query System. A subset of the Structured English Query Language (SEQUEL, or SQL) developed by IBM. Linked files, stored views, and nested queries result in a complete query capability. File system interaction isolated in an interface module. Extensive documentation guides user development of interfaces to other record oriented file handlers.

Portable Application Support System

Portable Windowing System. Hardware independent windowing system with borders, attributes, horizontal and vertical scrolling. User can construct interface file for any hardware. Interfaces provided for PC/XT/AT (screen memory interface and BIOS only interface), MS-DOS generic (using ANSI.SYS), Xenix (both with and without using the curses interface), and C-library (no attributes).

Screen I/O, Report, and Form Generation Systems. Field level interface between application programs, the Query System, and the file system. Complete input/output formatting and control, automatic scrolling on screens and automatic pagination on forms, process intervention points. Seven field types: 8-bit unsigned binary, 16 bit signed binary, 16 bit unsigned binary, 32 bit signed binary, monetary (based on 32 bit binary), string, and date.

Including Source Code

$395.00

File System interfaces include C-tree and BTRIEVE.

HARDWARE AND FILE SYSTEM INDEPENDENT

KURTZBERG COMPUTER SYSTEMS

41-19 BELL BLVD.
BAYSIDE, N.Y. 11361

VISA/Master Charge accepted
(718) 229-4540

*C-tree is a trademark of FairCom

IBM, SEQUEL, PC, XT, AT are trademarks of IBM Corp.
MS-DOS and Xenix are trademarks of Microsoft Corp.

CQL and the CQL Logo are trademarks of Kurtzberg Computer Systems.

CIRCLE NO. 148 ON READER SERVICE CARD

A Hard Look at LAN Choices.

Novell's LAN Report Package makes choices easier.

The flexibility of local area networks allows users to assemble LANs using network components that best suit the needs of the installation. But choosing those components can be a confusing process.

Novell, Inc., has published two reports designed to make the process easier: the *LAN Operating System Report 1986* and the *LAN Evaluation Report 1986.*

These reports help users evaluate network components and make informed decisions when choosing the components that meet their needs. Hardware and software issues are separately evaluated in the two reports, and extensive performance benchmarks are included.

Software Choices.

Choosing a network operating system, or LAN software, is the most critical aspect of designing a network. Simply, the better the operating system, the better the network. The *LAN Operating System Report* contains an in-depth analysis of LAN software, beginning with an examination of LAN software standards such as MS-DOS 3.1 and NETBIOS, and the file server environment. Issues like internetworking, system reliability, security and performance are addressed as well.

The *LAN Operating System Report* also evaluates Novell Advanced NetWare, the IBM PC Network Program and 3Com 3+. The report shows users how the design and implementation of these products translates into real performance.

Hardware Options.

The *LAN Evaluation Report 1986* focuses on evaluating network hardware. It examines hardware issues that affect LAN performance, including an analysis and benchmarking of major LAN products.

"Hardware and software issues are separately evaluated in the two reports..."

A key element of the study is the NetWare Evaluation System. The system provides a mechanism for matching site needs to specific hardware. Whether a new network is being planned or an existing site is being upgraded, the study is useful in the performance evaluation of any network.

System planning starts with the network interface card (NIC) and cabling. NICs analyzed in the study are:

- AT&T StarLAN
- Corvus Omninet
- Davong MultiLink
- Gateway G-Net
- IBM PC Network
- IBM Token Ring
- Interactive Systems Vista LAN/PC
- Nestar PLAN 2000
- Novell S-Net
- Proteon ProNET
- Standard Microsystems ARCNET
- 3Com EtherLink
- 3Com EtherLink+

The report analyzes each NIC according to its access scheme, raw bit rate, on-board processor and NIC-to-host transfer method.

Another important component of the LAN is the network server. In examining network servers, the *LAN Evaluation Report* looks at several performance indicators. Processor type is the most obvious feature to differentiate servers. However, other factors important in determining server performance are also evaluated, including processor clock cycle speed, wait states, server memory cycle speed, memory channel and transfer bus channel. And the report examines the effect of disk channel speed on network performance.

In addition to providing a careful examination of LAN hardware, the *LAN Evaluation Report* features an evaluation formula. Using the formula, a LAN's estimated future site activity is measured and matched to the appropriate LAN hardware.

To Get the Reports.

The *LAN Operating System Report 1986* and the *LAN Evaluation Report 1986* are available free of charge from Novell. To obtain a copy of the Novell Report Package, call or write Novell Corporate Communications, 122 East 1700 South, Provo, Utah 84601, (801) 379-5900.

NOVELL

CIRCLE NO. 109 ON READER SERVICE CARD

RICHARD L. SCHWARTZ and ROBERT E. SHOSTAK

What Computers Cannot Do

Certain mathematical problems simply cannot be solved by computer, no matter how big the program or how much time is available.

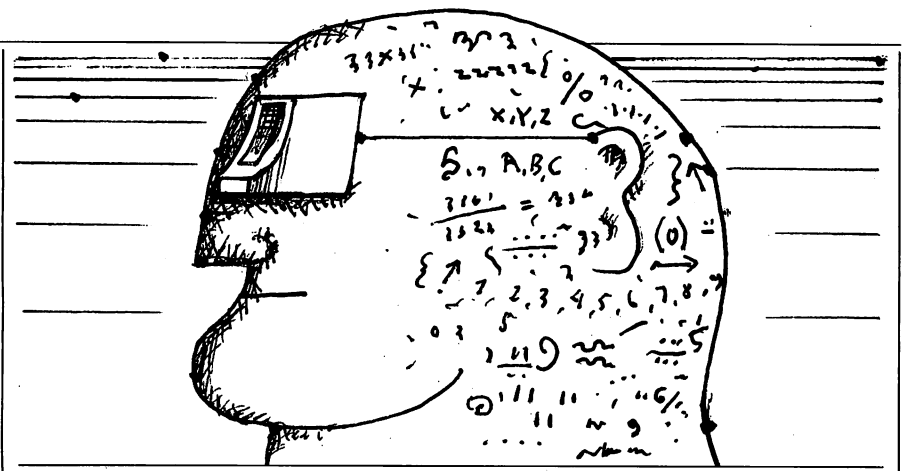


ILLUSTRATION • MACIEK ALBRECHT

Artificial intelligence is usually defined as a set of techniques for programming computers in a way that confers some of the attributes of human intelligence. The degree to which this endeavor will be successful is, of course, an open question. In principle, there is no reason why computers cannot be made as intelligent or more intelligent than humans—indeed, the human brain appears to be nothing more than a massively parallel personal computer based on some inexpensive organic logic. A much more interesting theoretical question is not whether computers can do what people do, but rather what is it that computers can compute, human or otherwise.

Ironically, the question of what types of problems can be solved purely through mechanical means was first posed long before the modern computer even came into existence. A great deal of attention was given to this issue in the 1920s and 1930s by a group headed by the well-known mathematician David Hilbert. While Hilbert and his colleagues had never heard of FORTRAN, they nevertheless understood the concept of a program, or algorithm. Their dream was to mechanize all of mathematics so that a single algorithm could solve any problem that could be stated in a mathematical way.

Hilbert and his associates made a certain degree of progress in the early 1930s by devising general algorithms that could solve certain classes of problems in mathematical logic. These early results were quite encouraging, and for a while it looked as if the livelihoods of unborn generations of computer programmers might be in jeopardy.

In 1936, however, hopes for implementing Hilbert's grand scheme were abruptly dashed by the logicians Alonzo Church and Alan Turing. Using different approaches, they proved that the so-called universal algorithm cannot exist. In fact, a consequence of what they demonstrated is even more remarkable: certain classes of problems in mathematics simply cannot be solved with computer programs, no matter how big the program or how much computer time is available.

Their arguments rest on a certain assumption that has come to be known as *Church's Thesis*, which, in essence, postulates that the instruction set of a very simple hypothetical computer known as the *Turing Machine* is powerful enough to use in coding any algorithm that could ever be programmed on any computer, existing or future.

The Turing Machine consists of a tape transport with an infinite tape, a read/write head, and a finite-state controller. The tape is divided into individual cells, each of which can store one bit of information. The read/write head scans a single cell of the tape at a time. The transport has the ability to move the tape one cell to the left or right upon the command of the controller. As might be expected, the controller is, at any given time, in one of a certain number of states. One of these is designated as the Start state; another is known as the Halt state.

The operation of the machine is quite simple. At each step, the read/write head reads the currently scanned cell of the tape. On the basis of the current state and on what is read, the controller either writes a bit to the tape or moves the head left or right one cell. In either case, it transits to another (possibly the same) state. Execution begins in the Start state and terminates in the Halt state. The input to the machine is encoded as the initial configuration of the tape; the output is given by what remains on the tape when the machine enters the Halt state.

The operation of the controller can be summarized by a table (think of it as a program) that lists the state and head transitions. Figure 1, for example, shows a Turing Machine program that takes a positive integer as input, and multiplies it by two. An integer x is encoded on the tape as a sequence of $x + 1$ 1s, with 0s everywhere else. As shown in the figure, the states of the machine are designated q_0 through q_8, with q_0 as the Start state and q_8 as the Halt state. Each entry in the table is a quadruple specifying the current state, the value of the currently scanned cell, a write or move action, and the next state that is to be entered. For example, the quadruple $q_4 0 R q_5$ specifies that if the machine is currently in state q_4 and scans a 0, it should move one cell to the right and enter state q_5.

As an exercise, the operation of the machine can be simulated for an input

Marshal Pascal
Discount!

We've seen a lot of Pascals, but this one takes the blue ribbon. Produces code smaller and faster than optimized C compilers. *ISO-compatible.* Supports the *8087 in-line* (8087 code emulation option if you don't have the chip). *True relocatable linker* allows access to the Microsoft family of languages and assemblers. *Four memory models. Overlays. Variable-length strings. Structured constants and structured function values. Separate compilation of modules. Procedural parameters.* Powerful compile options (optimization by-pass for quicker compiles, syntax evaluator, I/O "fine tuning", etc.). *Turbo Pascal Translator* brings your present Borland programs over to a ISO/Marshal-readable format. Watch the difference it'll make in your software's code size and speed!

Suggested retail is $189.00.
Our price is $159.00.

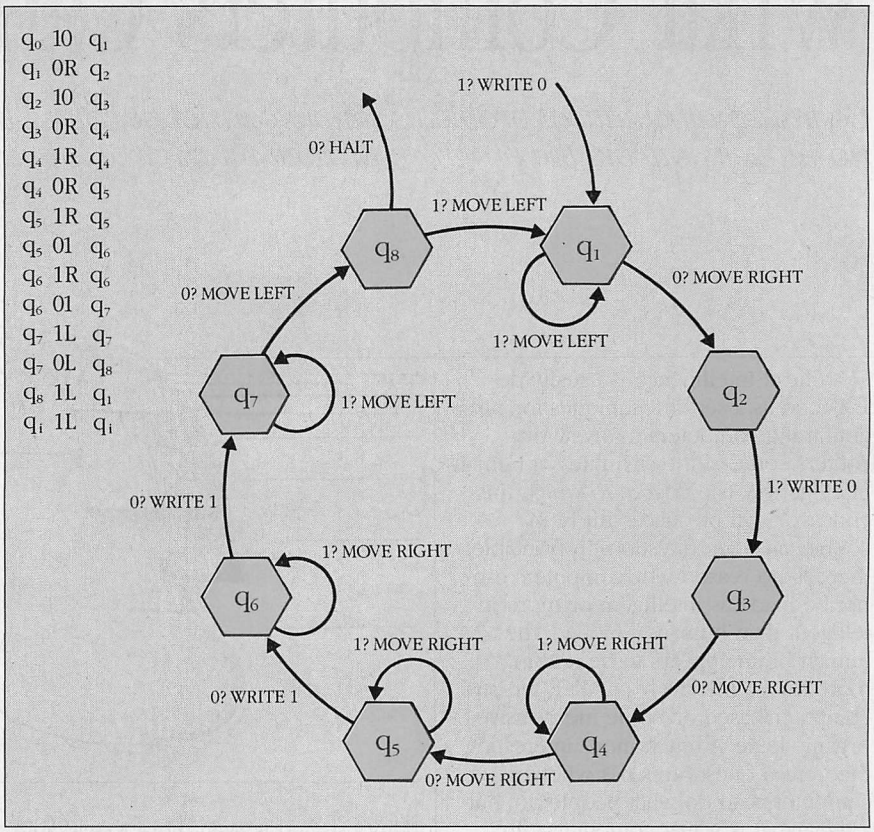
FREE OFFER! ABC WRITER, a powerful WordStar clone with full Print/Merge capabilities FREE with each copy of Marshal Pascal! Only while supply lasts.

Call (415) 930-9848—Ask for our free catalog of other software.

INNOVATION COMPUTERS
223 Donegal Way
Martinez, CA 94523

Turbo Pascal is a trademark of Borland International.
WordStar is a trademark of MicroPro Inc.
Microsoft is a trademark of Microsoft Corporation.

FIGURE 1: *Turing Machine Program*



Each entry is a quadruple specifying the current state, the value of the currently scanned cell, a write or move action, and the next state to enter.

of 2 (input tape ...01110...). Assume that the read/write head is initially positioned at the first 1. The quadruples can be grouped into subroutines for performing actions, such as erasing an input digit or adding new output digits.

With a little effort, a Turing Machine program can be written to multiply two numbers. More remarkably, and with quite a bit of effort, a Turing Machine can be programmed to emulate the operation of a Cray supercomputer. In fact, because a Turing Machine has an infinite tape, while the Cray's secondary storage is finite (at least, as it comes from the factory), the argument could be made that the Turing Machine is actually more powerful than a Cray. The important point, however, is that even if the Cray had an infinite secondary storage, it could not compute anything that could not be computed, given enough patience, using a simple Turing Machine.

Church's Thesis actually goes one step further. It says, in effect, that no matter what new hardware technology comes along, we will never be able to compute what we cannot compute now, given enough disk space and patience. Of course, this thesis is only a conjecture—it cannot be proved, and, in fact, it cannot even be stated in a precise, mathematical way. Still, it seems quite plausible, and at least so far, it has stood the test of time.

What Turing showed specifically was that it is not possible to come up with an algorithm that solves the so-called *halting problem*—determining whether a given computer program terminates (as opposed to looping forever) on all inputs. Imagine writing an assembly language program on the PC that reads a file containing an arbitrary Pascal procedure and determines if there is some input for which that procedure just loops forever without terminating. In fact, no such 8086 program exists—none could be written even if it could be arbitrarily large (but finite) and could take advantage of an infinite amount of disk space.

Of course, Turing had never heard of an 8086 or Pascal. He phrased the halting problem in terms of Turing Machines. How did Turing prove that the halting problem could not be solved? The answer is by using a clever technique known as *diagonalization.*

CIRCLE NO. 135 ON READER SERVICE CARD

Clarify your source code

C, BASIC, Pascal, dBASE, Modula-2 programmers: be more productive with two new utilities from Aldebaran Laboratories

"Occasionally, a utility comes along that makes a programmer's life much easier. SOURCE PRINT is such a program. It contributes to the programmer's job by organizing code into a legible format and by helping to organize the documentation and debugging process."

— PC Magazine
Sept. 16, 1986

Source Print and Tree Diagrammer both have easy-to-use menus with point-and-shoot file selection, and let you search for files containing a given string. For IBM PC and compatibles with 256K.

Join thousands of programmers who are working more efficiently using Source Print and Tree Diagrammer. Order these indispensable tools today. We ship immediately, and there's no risk with our 60-day money-back guarantee. **Order both and save. Only $115.00.**

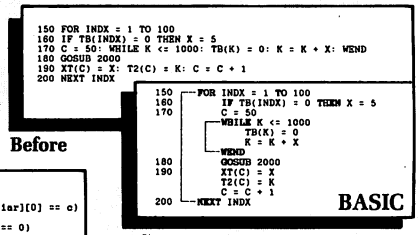
800-257-5773 Dept. 33
In California:
800-257-5774 Dept. 33

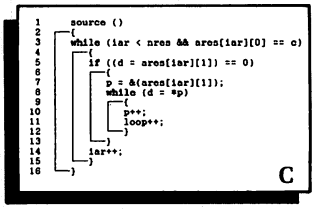
MasterCard, VISA, American Express, COD. Add $5 for shipping.

Source Print and Tree Diagrammer are trademarks of Aldebaran Labs. dBASE is a trademark of Ashton Tate.

Source Print™

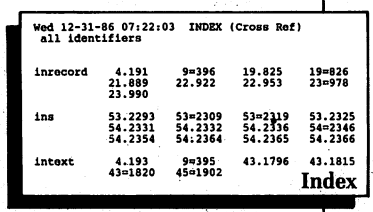
organizes your source code, simplifies debugging, and makes documentation a snap! It lists one or more source files with informative page headings and optional line numbers, while offering invaluable features:





The Index (Cross-Reference list) saves you time by showing exactly where variables are used and where functions, procedures, and routines are called.

$75.00

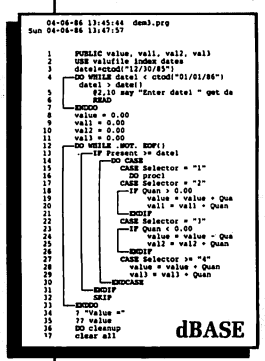


Locations where new values may be assigned to variables are shown, making it easy to track down that mysterious value change.

Structure Outlining solves the problem of hard-to-see nested control structures by automatically drawing lines around them.

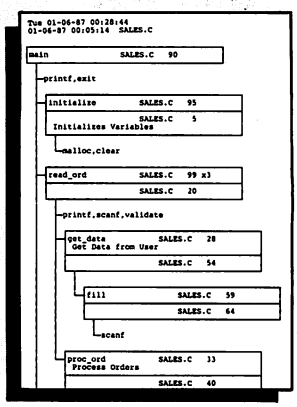
Automatic Indentation of source code and listings reduces your editing time and ensures indentation accuracy.

Plus . . . Source Print generates a table of contents listing functions and procedures. Keywords can be printed in boldface on most printers. Multi-statement BASIC lines can be split for readability. Functions and procedures can be drawn by name from one or more source files to form a new file.



Tree Diagrammer™

shows your program's overall organization at a glance. Ordinary program listings merely display functions, procedures, and subroutines sequentially, but do not identify the relationships between these routines. Our revolutionary new Tree Diagrammer automatically creates an "organization chart" of your program showing the hierarchy of calls to functions, procedures, and subroutines. Recursive calls are indicated and designated comments in the source code will appear on the chart.



Tree Diagrammer helps you organize your program more logically. And you'll be amazed at how easy it is to debug when you see how your routines interact.

$55.00

Aldebaran Laboratories 3339 Vincent Rd. Pleasant Hill, CA 94523 415-930-8966

YES! Rush me ☐ Source Print @ $75. _____ ☐ Tree Diagrammer @ $55. _____
☐ Both $115. Ship/Handling $5. For CA add 6% tax _____ Total _____
Name _____
Company _____
Address _____
City _____ State _____ Zip _____
☐ Check enclosed ☐ VISA ☐ MasterCard ☐ American Express
Card # _____ Exp. Date _____
Signature _____ Phone # _____ 3C

CIRCLE NO. 106 ON READER SERVICE CARD

dBASE II / III to C

the dBx™ translator

- **dBx** produces quality **C** direct from dBASE II or III programs.
- Move dBASE programs to UNIX or other machines.
- Improve program speed and reliability.
- Support multi-user/network applications.
- With power guidebook of conversion hints.
- Includes full screen handler and uses your current **C** database manager.
- May be used to move existing programs or help dBASE programmers learn **C** easily.
- For MSDOS, PCDOS, UNIX, XENIX, Macintosh, AMIGA. (Uses ANSI.SYS driver on MSDOS, CURSES under UNIX)
- Priced from $350, also available from distributors.

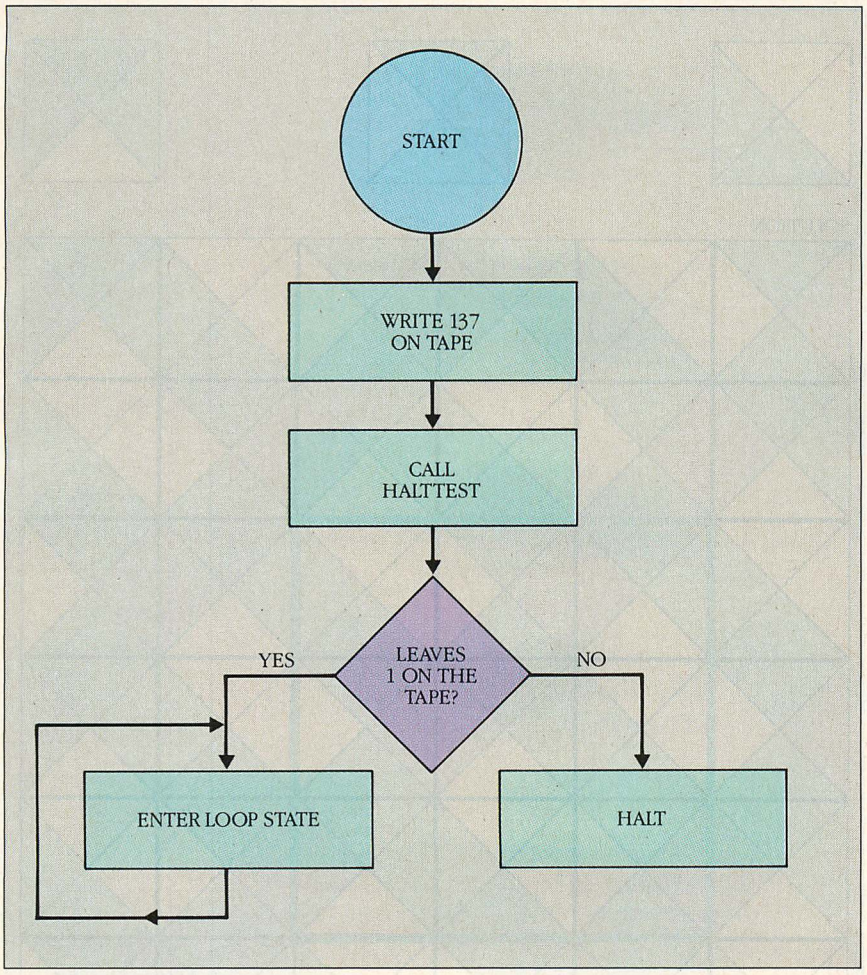
dBx is a trademark of **Desktop Ai**

1720 Post Road E., Westport, CT 06880 MCIMAIL • DESKTOPAI
Phone • 203•255•3400 Telex • 6502972226MCI

CIRCLE NO. 214 ON READER SERVICE CARD

386•386•386•

FIGURE 2: *Flowchart for KILLER*



Assuming that its Goedel number is 137, KILLER calls HALTTEST with 137 as an argument. If the call to HALTTEST leaves a 1, KILLER loops forever.

Here is the essence of the argument. First, in principle, all possible Turing Machine programs can be enumerated and assigned an integer. Naturally, an infinite number of such programs exists, but, nevertheless, no more than the number of available integers. Such an enumeration of all possible programs is called a *Goedel numbering*. Goedel numbering is the theoretician's version of the story about the comedians who tell the same jokes over and over so often that they simply assign each joke a number. (Someone calls out 45, and only one comedian laughs—he had not heard it before.)

Suppose that a Turing Machine program could solve the halting problem. This program (call it HALTTEST) could take as its input the Goedel number of an arbitrary Turing Machine program and determine whether or not that program halts on all inputs. If so, it would leave a 1 on the tape; if not, it would leave a 0.

Next, consider a powerful and easy-to-use program named KILLER that calls HALTTEST as a subroutine, as shown in figure 2. (To be completely rigorous, KILLER should be specified as a set of quadruples; the flowchart shown in the figure is an abbreviation.) Assuming that its Goedel number is 137, KILLER ignores its input and just calls HALTTEST with 137 as an argument. If the call to HALTTEST leaves a 1, KILLER loops forever; otherwise it quits, leaving a 0 on the tape.

Either way, a contradiction is apparent. Because HALTTEST halts on all inputs, KILLER hangs if and only if HALTTEST(137) leaves a 1 on the tape. But by definition of HALTTEST, HALTTEST(137) leaves a 1 on the tape if and only if KILLER never hangs. The conclusion, therefore, is that such a HALTTEST program cannot exist. This argument works just as well for 8086s as it does for Turing Machines. Because 8086s are so complicated to describe,

PROTECT YOUR COPIES OF PC TECH JOURNAL

Make your collection of PC TECH JOURNAL a handsome addition to your office or home—and protect and organize them for easy reference!

PC TECH JOURNAL Magazine Binders and cases are made of durable luxury-look leatherette over quality binder board. Custom designed for PC TECH JOURNAL, every order receives FREE transfer foil to mark dates and volume numbers.

FOR FAST SERVICE CALL TOLL-FREE 1-800-972-5858

MAGAZINE BINDERS

Hold your issues on individual snap-on rods. $8.95 each; 3 for $25.75; 6 for $48.75.



OPEN BACK CASES

Store your copies for individual reference. $7.95 each; 3 for $22.95; 6 for $43.95.



PC TECH JOURNAL
P.O. Box 5120
Philadelphia, PA 19141

Please send ☐ Binders ☐ Cases Quantity _____

Payment enclosed $_____.* Add $1 per order for postage & handling. (Outside USA, add $2.50 per unit ordered, US currency only.)

Charge my:
☐ Amex ☐ Visa ☐ MC (Minimum order $10.)

Card No._____ Exp. Date_____

Mr./Mrs./Ms._____
please print full name

Address_____

City_____

State_____ Zip_____

*PA residents add 6% sales tax.

THE SOURCE FOR ALL IBM PC EXPERTS.

There's one place to find the information about the sophisticated applications and products you need at your work place. PC TECH JOURNAL.

It's the magazine that brings you the in-depth coverage about the products and issues you have to know more about 13 times a year!

Guarantee delivery of the technical information and insights the systems experts of PC TECH JOURNAL deliver issue after issue and *save 50%!*

PC TECH JOURNAL P.O. Box 2996 Boulder, CO 80322

YES Send me PC TECH JOURNAL for:

☐ One year (13 issues) only $26.70.

☐ Two years for only $53.35.

SAVE 50%!

Savings based on annual single-copy price of $53.35.

Mr./Mrs./Ms._____

Company_____

Address_____

City_____ State_____ Zip_____

☐ Bill me ☐ Payment enclosed

Add $6 per year for postage outside USA, US currency only. Please allow up to 60 days for delivery of first issue.

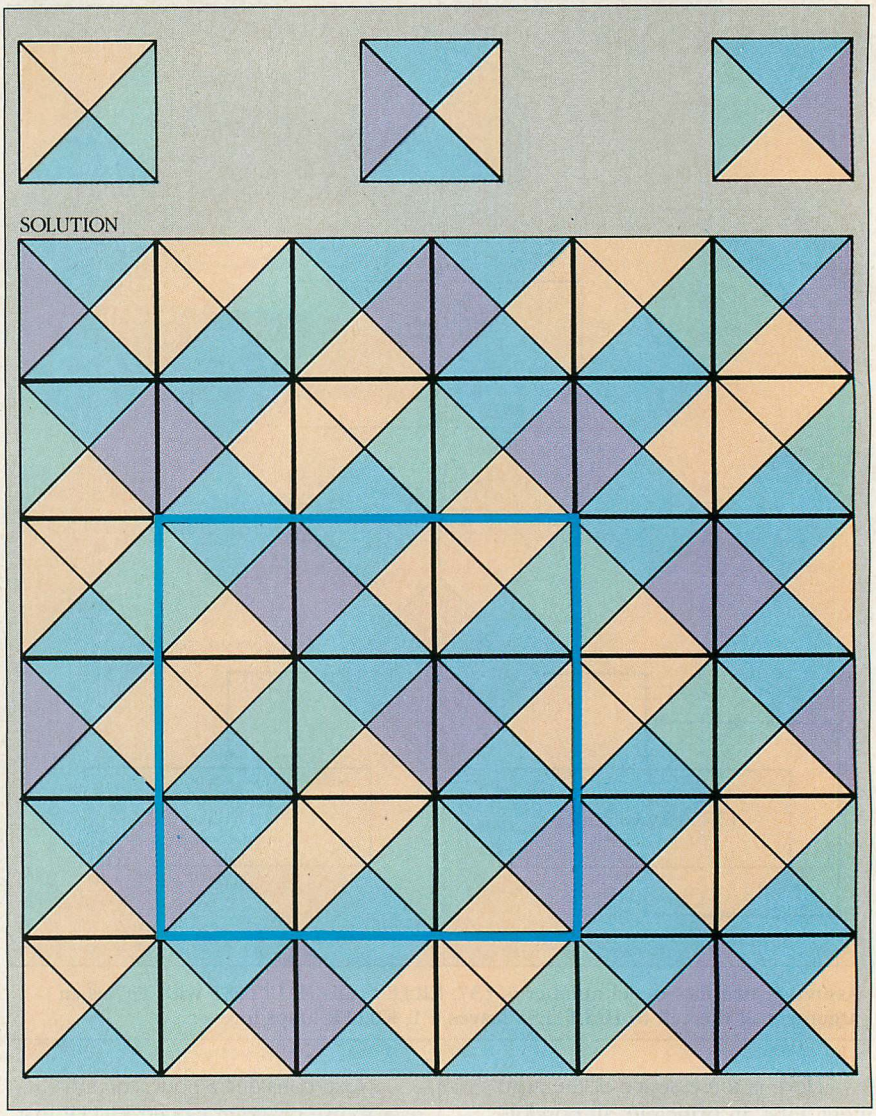
Annual basic subscription price is $34.97.

For faster service call Toll-Free 1-800-852-5200 today!

Your subscription includes the Special PC TECH JOURNAL Directory published in November!

4Z643

FIGURE 3: *Domino Example and Solution*



To solve this set of dominoes, a pattern is generated (inside blue square) whose left and right edges have the same color sequences, as do the top and bottom.

however, it is easier to give a rigorous formulation for Turing Machines.

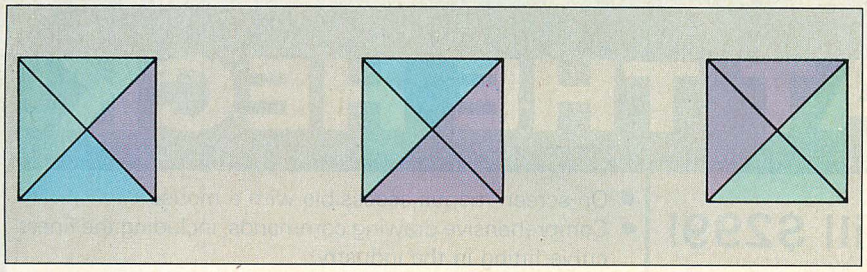
The halting problem is the prototypical instance of certain problems in theoretical computer science known as *decision problems*. An infinite class of yes-no questions is said to be *decidable* if and only if a computer program can be written that is guaranteed to answer each question in the class.

The *undecidability* of the halting problem has implications for fundamental questions in mathematics and logic. For example, the principle underlying this result is the basis of the well-known Goedel's Incompleteness Theorem, which states that certain mathematical truths simply cannot be proved. The consequences are not merely of theoretical interest, however. Believe it or not, the inability to solve the halting problem imposes limits on how well compilers can optimize compiled code. More generally, it limits the ability to write programs that can analyze other programs. It implies, for example, that programmers can never hope to write a single, general-purpose program that determines whether or not a piece of code contains bugs.

SOLVABLE DOMINOES

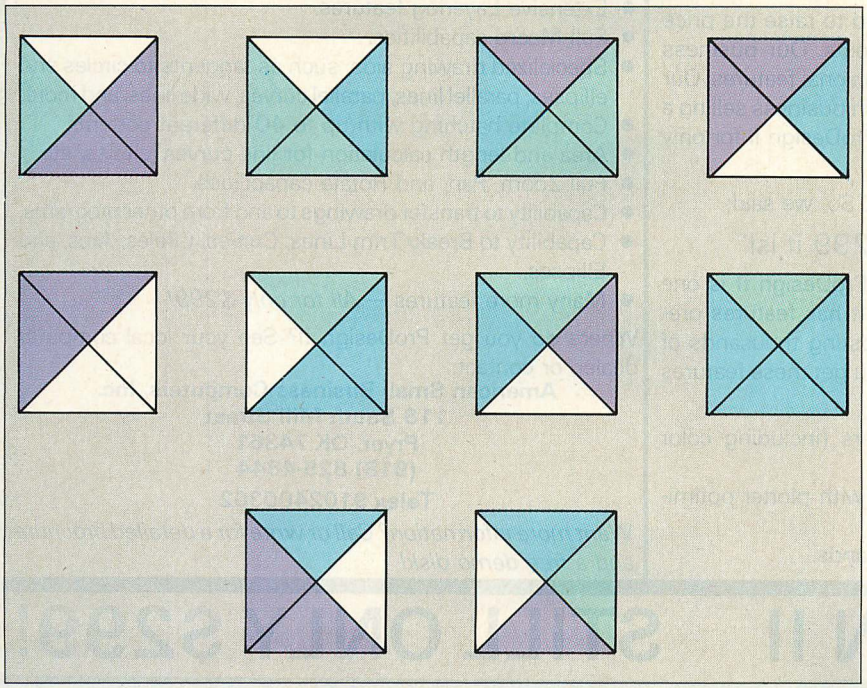
An interesting game was devised by Hao Wang at Bell Laboratories just for the purpose of studying the halting problem and other questions of decidability. Wang's game is a variation of solitaire played with colored tiles he refers to as *dominoes*. His dominoes are square-shaped and are painted in such

FIGURE 4: *Three-piece Domino Problem*



No general method exists for determining whether or not a given set of dominoes has a solution. This three-piece set may or may not be solvable.

FIGURE 5: *Ten-piece Domino Problem*



Even without using a general method, a player can still try ad hoc approaches to determine whether or not a particular set of dominoes can be solved.

a way that each of the four sides can be a different color. Moreover, each square has a definite orientation—that is, it cannot be rotated. In this domino game, players are given a finite set of dominoes of which they are allowed to make as many copies as they wish. The object of the game is to cover an infinite plane with dominoes in such a way that adjacent edges are the same color. If the entire plane can be covered in this way, then the set of dominoes is said to be *solvable*.

Figure 3 shows a set of three dominoes. This particular set is solvable, because it can be used to tile the plane using the pattern also shown in figure 3. The pattern is generated by repeating a nine-domino unit whose left and right edges have the same sequences of colors, and whose top and bottom edges also follow the same color sequences. Clearly, any time a repeatable unit of this type can be formed, a solvable set is possible. (It is possible to show, however, that some solutions to domino problems do not consist of a repeating block.)

Wang was able to use dominoes to simulate Turing Machine computations and therefore to create the equivalent of the halting problem. Wang and his colleagues showed that for a given Turing Machine program, a set of dominoes can be devised that has a solution if and only if that program hangs. His construction starts with the table of quadruples giving the program and generates dominoes that simulate the action of reading the tape, making state transitions, and so on. Because the halting problem is undecidable, it follows that no general method exists for determining whether a given set of dominoes has a solution.

Even without a general method, ad hoc approaches still can be used to determine whether or not a set of dominoes has a solution. Figures 4 and 5 present two sets of dominoes that may or may not be solvable. Solutions or arguments for unsolvability of these sets will be published in our next column.

While Wang's reduction is too complex to detail here, it should be noted that an important step along the way was the observation that a particular set of dominoes can tile the entire plane if and only if it can tile a single quadrant. Obviously, given that a player can tile the whole plane, he can cut out three of the four quadrants in order to find a solution for a quadrant. The converse is by no means obvious, but can be established by noting that a tiling for a quadrant necessarily contains a subtiling of size n-by-n, for each n. An infinite tree of these partial solutions can be constructed, and an infinite path can be found through the tree in order to yield a solution that covers the entire plane. Therefore, if a quadrant can be tiled, so can the plane.

Wang's games of dominoes were not merely of recreational interest. He was able to use them in order to close what was then an open question in mathematical logic. Specifically, he showed that the problem of determining the validity of members of a certain class of logical formulas can be simulated using dominoes. The formulas in question have the form "There exists an x such that for all y, there exists a z such that. . ." followed by a subformula containing no "for all" or "there exists."

Perhaps the greatest importance of Wang's dominoes is that they vividly illustrate an important technique used in many computer science disciplines and particularly in artificial intelligence—that is the technique of solving a problem by reducing it to another problem whose solution is already known. The domino reduction actually uses a contrapositive form of that technique, in which a given problem is shown to have no solution by reducing another problem (the Halting problem, in this case) that is known to be intractable to the given problem.

Richard Schwartz, Ph.D., and Robert Shostak, Ph.D., are vice presidents of software development and cofounders of Ansa Software.

AW . . .
WHAT THE HECK!

ProDesign II — Still $299!

ProDesign II — the Easy-to-Use CAD System. The CAD system that was introduced two years ago for the amazingly low price of $299. The CAD system that has undergone four major revisions with more than 400 enhancements. The CAD system that still costs only $299!

How do we do it? Our accountants said to raise the price to cover research and development costs. Our business advisors said to charge more for the additional features. Our competitors said it's impossible to stay in business selling a comprehensive CAD package such as ProDesign II for only $299.

And our customers said $299 is great. So, we said:

"Aw . . . What the Heck! $299 it is!"

Now, two years after its introduction, ProDesign II is one of the world's leading CAD packages. It has features previously found only on CAD systems costing thousands of dollars. For the single price of $299, you get these features and more:

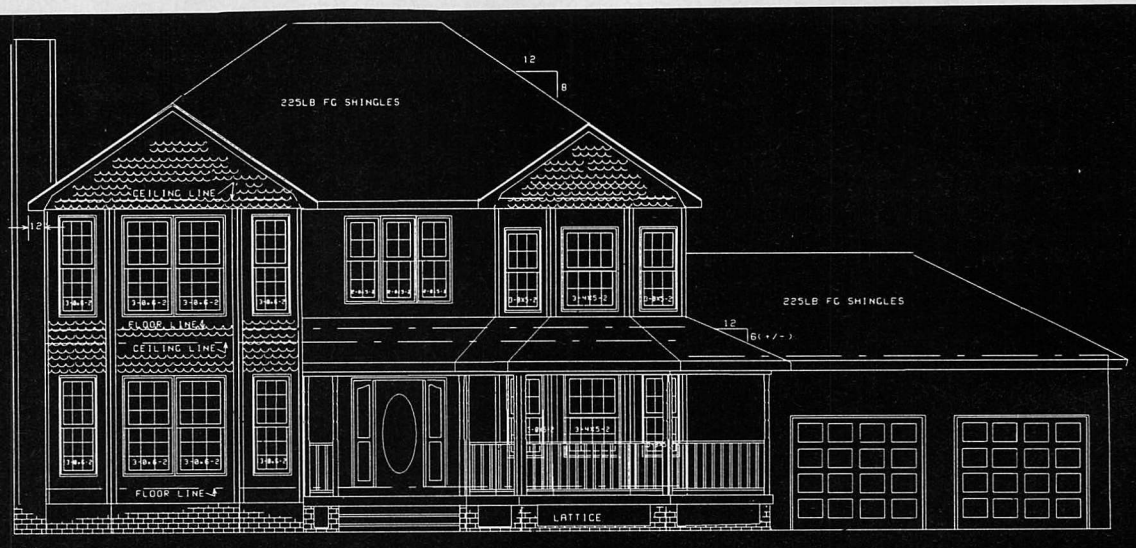
- Support for more than 180 printers (including color printers)
- Support for more than 80 plotters, with plotter optimization.
- Easy-to-Use single keystroke commands.
- On-screen menus accessible with a mouse.
- Comprehensive drawing commands, including the finest curve fitting in the industry.
- Editing features unsurpassed by ANY other CAD package.
- Extensive snap features, including snap to point, endpoint, midpoint, line, circle, ellipse, arc, intersection, and perpendicular.
- True Auto Dimensioning with several formats.
- Extensive Layering features.
- Full Macro capabilities.
- Specialized drawing aids, such as tangents to circles and ellipses, parallel lines, parallel curves, wide lines, and more.
- Complete hatching with up to 40 different patterns.
- Area and length calculation for line curves, circles, etc.
- Full Zoom, Pan, and Rotate capabilities.
- Capability to transfer drawings to and from other programs.
- Capability to Break/Trim Lines, Curves, Circles, Arcs, and Ellipses.
- Many more features — *All for only $299!*

Where do you get ProDesign II? See your local computer dealer, or contact:

American Small Business Computers, Inc.
118 South Mill Street
Pryor, OK 74361
(918) 825-4844
Telex 9102400302

Want more information? Call or write for a detailed brochure and a free demo disk!

PRODESIGN II STILL ONLY $299!



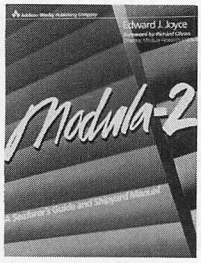
FRONT ELEVATION

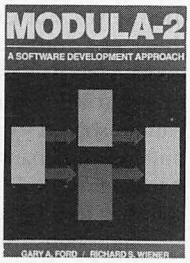
1/4" = 1'-0"

CIRCLE NO. 118 ON READER SERVICE CARD

Approaching Modula-2

Two books attempt to complement Nicklaus Wirth's own work regarding his Modula-2 programming language.





Modula-2: A Seafarer's Guide and Shipyard Manual
Edward J. Joyce (Addison-Wesley; Reading, MA; 1985) 270 pages; paper, $16.30

Modula-2: A Software Development Approach
Gary A. Ford and Richard S. Wiener (John Wiley & Sons; New York, NY, 1985) 400 pages; paper, $30.75

The programming language Modula-2 is gathering a large, and ever-growing, following in the computing community. Several good PC implementations of the language have recently appeared, making Modula-2 a strong contender for use in any PC programming project (see "Modular Developments," John Cockerham, this issue, p. 114). The two books reviewed here take differing viewpoints in their approaches to this modular programming language.

What is needed most of all on a Modula-2 programmer's bookshelf is a complete, accurate, easy-to-understand reference book on this modern descendant of Pascal. Edward J. Joyce's *Modula-2: A Seafarer's Guide and Shipyard Manual* goes only partway toward achieving this goal. Its relaxed, conversational style provides welcome relief from the dry, formal writing of the definitive Modula-2 text, *Programming in Modula-2* by Niklaus Wirth, the original designer of the language. Wirth's book, which is filled with Backus-Naur notation and cryptic writing, is almost too concise to be readable.

On the other hand, *A Seafarer's Guide*, while much more readable, sacrifices accuracy and completeness. The organization of the book is, indeed, quite baffling; by chapter 2, the author has presented several sample programs, each of which uses features of the language that have not yet been explained. For example, Joyce discusses the formatted input and output available through the Modula-2 library procedures WriteInt and ReadInt despite the fact that these procedures are not examined in detail until 70 pages later.

One of Modula-2's most important features is its ability to handle separate compilation. It is this capability that makes the language attractive for large, team programming projects and for any software development effort in which data and procedural abstraction and information hiding could be useful. If Modula-2 is nothing else, it is the *modular* programming *language*. Yet, in this book, modules and separate compilation are glossed over until the last 30 pages of the text, where they are given, at best, a cursory treatment. The author places little emphasis on the features that make Modula-2 an interesting and unique programming language, and chooses instead to arrange the text around the syntax alone.

Sample Modula-2 code appears throughout the text, but few complete programs are listed, and those that are complete, are simplistic. One 16-line program is the only example that is given in the text of Modula-2's separate compilation facilities.

The book does not present a cohesive picture of the standard Modula-2 function libraries for file and terminal I/O, dynamic memory management, real math functions, and so on. These libraries are not guaranteed to be identical across implementations, so it is rather important to distinguish between base language features and features of the library modules. This important distinction is left unclear in the text because base language and library features are scattered throughout the book and are only explained as they crop up in the sample programs.

Joyce's *Modula-2: A Seafarer's Guide and Shipyard Manual* is an appropriate text for the programmer who simply wishes to learn the syntax of Modula-2 quickly and painlessly. It is not well suited for the software developer who is interested in understanding the reasoning behind the design of Modula-2, nor is it appropriate for the programmer who is evaluating Modula-2 as a possible implementation language for a software project. Its haphazard organization, poor examples, and lack of emphasis on the strong points of the language make this a book that is truly lost at sea.

Modula-2: A Software Development Approach, by Gary A. Ford and Richard S. Wiener, takes a completely different tack and, as a result, offers the reader a more complete treatment of its subject than does Joyce's book. *A Software Development Approach* is more a text on the methods of software engineering than a guide to a particular programming language. Although the book is divided into two sections—the first part is on Modula-2 as a language, and the second is on Modula-2 as a tool for software development—the emphasis of the entire book is, as its title suggests, on software development.

As the authors present each feature of Modula-2, they are careful to discuss not only its syntax, but also the reason for including it in the language; the reader is not left to guess the intentions of the language designer.

Ford and Wiener present many relevant examples, and the sample programs illustrate well both the syntax of the language and the software development concepts (data and procedure abstraction, information hiding, and so on) that Modula-2 was, in fact, designed

THIS CARD GIVES A COLLEAGUE A YEAR'S WORTH OF VALUABLE INFORMATION AND INSIGHT



When you give PC TECH JOURNAL as a gift to a client or business colleague, you're giving valuable information on systems design and integration.

PC TECH JOURNAL is the recognized source of expert facts for designers, integrators and DP/MIS professionals who require current information and sophisticated applications and input.

Give a gift that's delivered 13 times* a year and save 50% off the annual single-copy price of $53.35.

*Your gift subscription includes the special PC TECH JOURNAL Directory issue published in November!

TECH JOURNAL
P.O. BOX 2966
BOULDER, CO 80322

4Z668

Send Gift TO:

Mr./Mrs./Ms. _____
please print full name

Company _____

Address _____

City _____ State _____ Zip _____

Send Card FROM:

Mr./Mrs./Ms. _____

Company _____

Address _____

City _____ State _____ Zip _____

☐ Bill me ☐ Charge my credit card for $26.70 for 13 issues.

Check one: ☐ AmEx ☐ Visa ☐ MasterCard

Card No. _____ Exp. Date _____

Add $6 for postage outside USA, US currency only. Please allow up to 60 days for delivery of first issue.

Basic annual subscription price is $34.97.

FOR FASTER SERVICE CALL TOLL-FREE 1-800-852-5200!

to support. Complete listings of several Modula-2 utility libraries (modules) are included; among them are complex number, stack, queue, and character string modules, each of which is implemented several different ways. Also listed are modules implementing dynamic memory allocation, process synchronization, and a simple text editor with a spelling checker.

The book mentions several shortcomings of the language (such as its lack of support for generic data abstraction), but emphasizes possible solutions and work-arounds, not the problems themselves. For example, when noting the very primitive nature of Modula-2's concurrent processing facilities, the authors provide listings of modules to handle shared variables, interprocess buffers, and channels, giving an outstanding example of Modula-2's flair for software tool building.

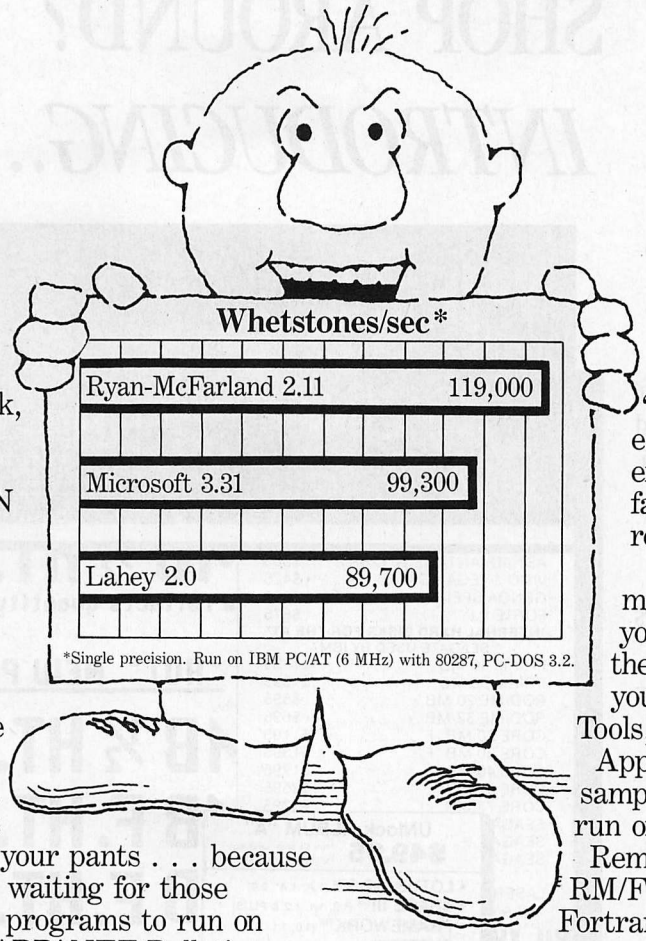
Although *A Software Development Approach* is concise and well-organized, it sometimes goes into more detail than is really necessary. For example, as they discuss dynamically-allocated data structures, Ford and Wiener explain the difference that exist between singly and doubly linked lists; this is not the material of a programming language text or even a text on software development, but rather one on basic data structures and algorithms. Wirth's own book on algorithms and data structures in Modula-2 was reviewed in "A Classic Revised," Book Reviews, Michael A. Covington, January 1987, p. 187.

This book is not ideal for those programmers who know Pascal and want to get up to speed quickly in Modula-2. However, such programmers should be able to skim the early chapters in order to understand the differences between Modula-2 and its predecessor, Pascal. Unfortunately, this text does not provide a summary of the differences between the two languages.

A Software Development Approach emphasizes software development so extensively that it would not be difficult to substitute another language, say Ada, for Modula-2 in the text. This is not to say that the book's description of Modula-2 is not accurate and complete; in fact, it makes a compelling argument for Modula-2 largely because it emphasizes principles of good software engineering. This book is a good reference for the serious Modula-2 user and an excellent introduction to the language for anyone considering using Modula-2 for a programming project.

—ARTHUR A. GLECKER

KILLER FORTRAN.



This is the one.

Whatever the yardstick, RM/FORTRAN™ blows the others away.

Sieve? RM/FORTRAN runs 26% to 228% faster.

Whetstones? RM/FORTRAN is 20% to 33% better.

And those aren't our benchmarks. They're the industry standards.

With RM/FORTRAN, you can actually feel the difference in the seat of your pants . . . because you'll do less sitting and waiting for those big, complex mainframe programs to run on your micro. In fact, an ARPANET Bulletin Board user said, ". . . for serious conversion work of mainframe Fortran code, use the Ryan-McFarland . . ."

RM/FORTRAN is also the first and only PC Fortran GSA certified at Full Level ANSI 77 with no discrepancies. So programs move effortlessly from your mainframe to a PC, XT, AT or compatible. And they also port to 68000-based systems or new 80386 machines. You may already be using version 1.0 under the name "IBM PC Professional Fortran by Ryan-McFarland."

RM/FORTRAN comes with popular extensions from VAX, VS and Fortran 66. That's more mainline mainframe extensions than any other micro Fortran.

And there's more. PC TECH Journal, for example, said our "compiler's documentation, ease of use, speed of execution, and debugging facilities place it first for recommendation."

But why spend any more time reading when you should be filling out the coupon and getting your free "RM/FORTRAN Tools, Utilities and Applications Directory" and sample benchmark programs to run on your own PC.

Remember, if it isn't RM/FORTRAN, it's just a Fortran also-ran.

Ryan-McFarland
609 Deep Valley Drive
Rolling Hills Estates, CA 90274
or call 213-541-4828

RYAN-McFARLAND

PCT-3

Knock me out. Send me the benchmark programs and the free Applications Directory.

Name _____

Company _____ Title _____

Street _____

City _____ State _____ Zip _____

Phone ( ) _____

CIRCLE NO. 137 ON READER SERVICE CARD

NEED IT FAST? WANT TO KNOW MORE? DON'T HAVE TIME TO SHOP AROUND?

INTRODUCING...

If you're like most system professionals you're up-to-date about the products in the PC marketplace. You're aware of brand and model differences, are informed about connectivity and compatibility problems, and you shop for competitive prices and fast service.

You're also probably among the many PC TECH JOURNAL readers who purchase by mail. That's why we're starting THE MART—PC TECH JOURNAL'S First Class Mail-Order Section.

Starting this month, and every month hereafter, you'll find the products you're looking for advertised in THE MART—and you'll benefit from the fast service and helpful support that identifies PC TECH JOURNAL advertisers.

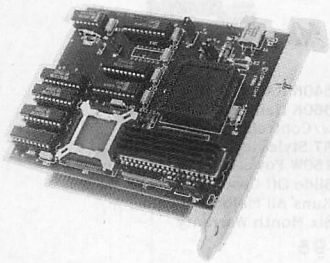
If you're ready for First Class service, you're ready for THE MART.



MICROWAY MEANS 8087 PERFORMANCE

FastCACHE-286™

Runs the 80286 at 8.5 or 11 MHz and the 80287 at 5, 6 or 11 MHz. Includes 8 kbytes of 55ns CACHE. Works with more PCs than any other accelerator, including Leading Edge Model D, Compaq, and Turbo motherboards. Includes 8088 Reboot Switch, DCache and Diagnostics..... **From $449**



LOTUS/INTEL EMS SPECIFICATION BOARDS

MegaPage™ The only EMS board which comes populated with two megabytes of cool-running, low power drain CMOS RAM installed. Includes RAM disk, print spooler, disk cache and EMS drivers. For the IBM PC, XT and compatibles...**$549**

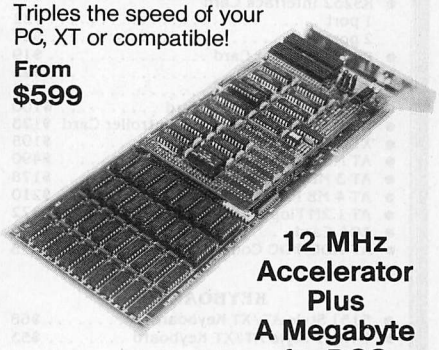
MegaPage with ØK..... **$149**

MegaPage AT/ECC™ EMS card for the PC AT and compatibles includes Error Correction Circuitry. With ECC, 11 RAM chips cover 256K so the user never encounters RAM errors. Sold populated with 1 megabyte CMOS . . . **$699** or with 3 megabytes CMOS cool running low power drain RAM . . . **$1295**. Optional serial/parallel daughterboard..... **$95**

NUMBER SMASHER/ECM™

Triples the speed of your PC, XT or compatible!

From $599



12 MHz Accelerator Plus A Megabyte for DOS

***PC Magazine* "Editor's Choice"**

DATA ACQUISITION and REAL TIME TOOLS

DAL™ – "Data Acquisition Language."

Unkelscope™ – A real time data acquisition, control and process software pkg.

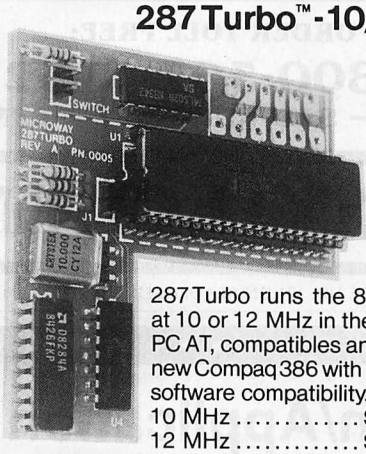
87FFT and 87FFT-2

TransView Menu driven FFT Spectrum/transfer analyzer..... **$250**

RTOS – REAL TIME OPERATING SYSTEM
A multi-user, multi-tasking real time operating system. Includes a configured version of Intel's iRMX-86, LINK-86, LOC-86, LIB-86, OH-86 and the MicroWay 87DEBUG. Runs on the IBM-PC, XT, PC-AT and COMPAQ..... **$600**

INTEL COMPILERS Available for RTOS
FORTRAN-86, PASCAL-86, PL/M-86.

287 Turbo™-10/12



287 Turbo runs the 80287 at 10 or 12 MHz in the IBM PC AT, compatibles and the new Compaq 386 with 100% software compatibility.
10 MHz **$450**
12 MHz **$550**

***PC Magazine* "Editor's Choice"**

8087 UPGRADES

All MicroWay 8087s include a one year warranty, complete MicroWay Test Program and installation instructions.

8087 5 MHz..... **$114**
For the IBM PC, XT and compatibles

8087-2 8 MHz..... **$149**
For Wang, AT&T, DeskPro, NEC, Leading Edge

80287-3 5 MHz..... **$179**
For the IBM PC AT and 286 compatibles

80287-6 6 MHz..... **$229**
For 8 MHz AT compatibles

80287-8 8 MHz..... **$259**
For the 8 MHz 80286 accelerator cards

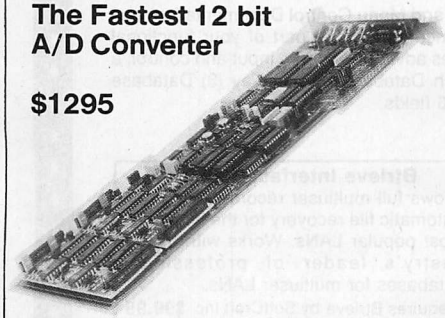
80287-10 10 MHz..... **$395**
For the Compaq 386

Call for prices on V20, V30, 64K, 128K and 256K RAM

A2D-160™

The Fastest 12 bit A/D Converter

$1295



160,000 Samples per second
Pseudo Random Noise Generator/DAC
Optional signal conditioners
AFM-50 Programmable Low Pass Filter Module..... **$225**

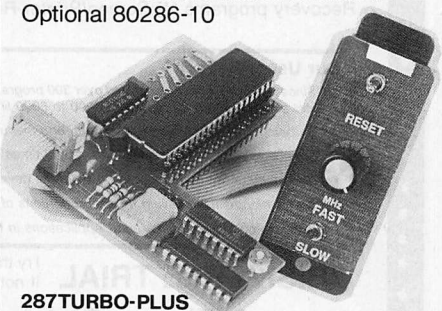
8087 SOFTWARE

IBM BASIC COMPILER	$465
MICROSOFT QUICK BASIC	$79
87BASIC COMPILER PATCH	$150
IBM MACRO ASSEMBLER	$155
MS MACRO ASSEMBLER	$99
87MACRO/DEBUG	$200
MICROSOFT FORTRAN	$209
RM FORTRAN	$399
LAHEY FORTRAN F77L	$477
MS or LATTICE C	CALL
STSC APL★PLUS/PC	$450
STSC STATGRAPHICS	$675
SPSS/PC+	$675
87SFL Scientific Functions	$250
PHOENIX PRODUCTS	CALL
FASTBREAK for 1-2-3 V.1A	$79
HOTLINK for 1-2-3 V.1A	$99

287 TURBO-PLUS™

Speeds up your AT

Adjustable 80286 Clock 6-12 MHz
10 MHz 80287 Clock
Plus Full Hardware Reset..... **$149**
Optional 80286-10



287TURBO-PLUS
With 80287 10 MHz..... **$549**
With 80287 12 MHz..... **$629**

CALL (617) 746-7341 FOR OUR COMPLETE CATALOG

MicroWay
P.O. Box 79
Kingston, Mass.
02364 USA
(617) 746-7341

The World Leader in 8087 Support!

MicroWay Europe
32 High Street
Kingston-Upon-Thames
Surrey England KT1 1HL
Telephone: 01-541-5466

PC XT/AT ADD-ON BOARDS

- MS DOS/GW Basic 3.2 ..... $80
- XT Mother Board/Bios $83
- XT Turbo Board/Bios $103
- Monochrome/Graphic/Printer Card ... $66
- Monochrome Graphic Card $60
- Color Graphic/Printer Card $60
- Color Graphic Card $50
- XT Multi I/O Card $73
- XT I/O Plus II $45
- Floppy Disk Controller
  - 1 port $25
  - 2 port $33
- RS232 Interface Card
  - 1 port $22
  - 2 port $27
- Parallel Printer Card $19
- Clock Card $25
- Game Card $18
- Hard Disk Controller Card $100
- Hard Disk/Floppy Disk Controller Card $123
- XT 2 MB RAM Card $195
- AT Mother Board/Bios $490
- AT 3 MB Multifunction Card $178
- AT 4 MB RAM Card $210
- AT 1.2M Floppy Disk Card $72
- EGA Card $245
- AT HDC/FDC Controller w/Cable $215

KEYBOARDS

- 5151 Style AT/XT Keyboard $68
- 747 AT Style AT/XT Keyboard $53

POWER SUPPLY

- 150 Watt XT Power Supply $53
- 200 Watt AT Power Supply $85

MONITORS



PARCO (Sony) Height Resolution 12" 90° Monitor 800 x 700 Lines With Non-Glare Screen/Swivels Base Amber. $115

- SAMSUNG — Amber $79
- TAXAN 620 Color $375
- TAXAN 630 Super Hi-Res. Color $445
- TAXAN 640 Super Hi-Res. Color $495
- TAXAN 760 EGA Monitor $499

PRINTERS

- RITEMAN PLUS (120 cps. 80 col.) $175
- RITEMAN - 15 (160 cps. 136 col.) $345
- BROTHER M1509 (180 cps. 136 col.) ... $395

DRIVES

- TEAC 360K Floppy Drive $90
- FUJISU 360K Floppy Drive $82
- CHINON 360K Floppy Drive $85
- 20MB Hard Disk/WD ..... $Call
- 30MB Hard Disk/WD ..... $Call
- 1.2MB TEAC AT Drive $135

CHASSIS

- Flip Top XT Case $29
- Slide Off XT Case $36
- AT Jr. Style XT Case $37
- AT Case $85

All Cases Include Speaker/Hardware

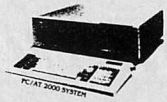
MODEMS

- Internal Modem-Everex $137

Select 300/1200 bps, powerful BitCom Communication Software included. Auto answer/dial.

- External Modem-Smarteam $160

PC/AT 2000 SYSTEM



- 80286 Processor (6/8MHZ)
- 1024K RAM
- 1.2MB Floppy Disk Drive
- AT Hard Disk/Floppy Disk Controller Card
- Clock/Calendar with Batter Backup
- AT Style Keyboard
- 200W Power Supply/Case
- Runs All Major Software
- Six Month Warranty

$1249

PC/XT 2000 SYSTEM



- 640K RAM
- 360K Half Height Floppy Drive w/Controller
- AT Style Keyboard
- 150W Power Supply
- Slide Off Case
- Runs All Major Software
- Six Month Warranty

$495

(201) 944-5002
2142 N. Hudson St.
Fort Lee, NJ 07024

IBM PC, IBM XT and IBM AT are trademarks OF IBM corpration.
MS-DOS is a trademark of Microsoft Corporation.

Prices Subject to Change without Notice.

ORDER TOLL FREE:

1-800-367-1132

MONDAY — SATURDAY 9AM — 6PM EST.

Customer Service
(201) 944-5010
9AM — 5PM EST. M-F

ORDERS SHIPPED UPS COD WITHIN 24 HRS.

SUNNYTECH INC.

CIRCLE NO. 184 ON READER SERVICE CARD

Turbo Screen/Application Generator

Be 3-6 times more Productive!!!

Guaranteed* For $69.95 (one month holiday special)

Turbo Master helps you develop your functional specs (Generates Screen, File, Isam, Variable and Menu Control Documentation) and then allows you to "Quickly" prototype a validation model of your system. (Which can be incorporated as part of your functional specifications.) Turbo Master can then generate a super-fast Turbo Pascal Program that features advanced screen input and control, a professional control menu, the database functions of (1) Add/Edit/Delete Records (2) Search Database by any Key (3) Database Recovery programs (4) Screen/Printer Report for each of the keys. Each Key can have up to 6 fields.

Our Users Report

- *"Since Fall of 85, I have generated over 300 program modules with it and find it to be just what I needed. Most all of the modules represent 5000 to 8000 lines of Pascal Code"* Oner Systems.
- *"By being able to produce a 21 screen and menu control demo so quickly helped me obtain the contract."*
- *"Speeded up my screen development by 6 times"* Elexor Associates.
- *"Has many of the features of the Super Mini development tools costing $10,000."* Applied Micro Systems.
- *"Saved months from having to recode portions of our system."* Real Green Inc.
- *"We developed 3 Vertical Market Applications in the 6 months we had your system."* Absolute Systems.

***RISK FREE TRIAL** Try the demo package included for 30 days. If not pleased return for a full refund.

Receive 6 Floppy Disks and a manual containing:

1. **Screen Painter/Editor & Generator** • Paint menu screens using keyboard • Has variable dictionary to provide consistant edits • Date entry masks • Date & range checks • Field and/or global help screens • Box & line drawing • Error & message handler
2. **Help Screen Maker** - Different help screen for each field.
3. **Menu Editor & Generator** • Allows selection by 4 methods.
4. **Database Program Generator** • Produces "Easy to Read" code that can be easily modified by experienced developers.
5. **Resident Isam Module** - compatible with Turbo Toolbox, but saves 8K of codespace and 10K of dataspace.
6. **Turbo Resident Screen Capture Utility** which allows you to capture Text Screens from any running program.

& Much, Much More

NO ROYALTIES on Generated Programs

Credit Card & C.O.D. Orders Call: 1-800-821-9503

In Florida 1-800-342-0137

Btrieve Interface Module

Allows full multiuser record locking and Automatic file recovery for the industry's most popular LANs. Works with the industry's leader of professional databases for multiuser LANs.
Requires Btrieve by SoftCraft Inc. **$99.95**

- ☐ Turbo Master by Hawaiian Village Software . . . $69.95
- ☐ Btrieve Interface by Innovative Interfaces 99.95
- ☐ Turbo Pascal by Borland International $99.95

For Further Information Call:
(305) 892-5686

Add 7.50 shipping to all U.S. Cities. All foreign orders add 15.00 per product ordered

CIRCLE NO. 132 ON READER SERVICE CARD

Btrieve is a trademark of SoftCraft Inc. Turbo Pascal & Turbo Database Toolbox are trademarks of Borland International.

Today's 386 compatibles fall into two categories:

Those that "enhance" technology.

The PC Designs GV-386 is not an enhancement. It's the fastest 80386-based compatible you can buy.

That's because the engineers here at PC Designs realized early on that to take full advantage of the 80386 chip, an equally revolutionary motherboard was needed.

So they started with a blank piece of paper and—from the ground up—designed a 80386 micro that blows the doors off every other compatible available.

Get Power Hungry

At the heart of the GV-386 is a 32-bit processor operating at 16MHz clock speed; keyboard switchable to 8MHz (it will even support 24MHz operation, once Intel makes that available).

And that's Zero Wait State clock speed, thanks to a specially-designed, high-speed memory cache circuit. When enabled, this unique circuit—with a full 64K of static RAM—allows you virtually instant data retrieval 80 percent of the time.

The cache circuit—a PC Designs exclusive—also ensures rock-solid reliability because it eases the load high speed puts on the integrated circuits. Even at zero wait state, the GV-386 never exceeds IC design specifications.

Radical, but compatible

Despite its radical innovations, the GV-386 was designed to retain the standard 8MHz IBM PC-AT bus timing. The result is unparalleled compatibility with existing software and hardware.

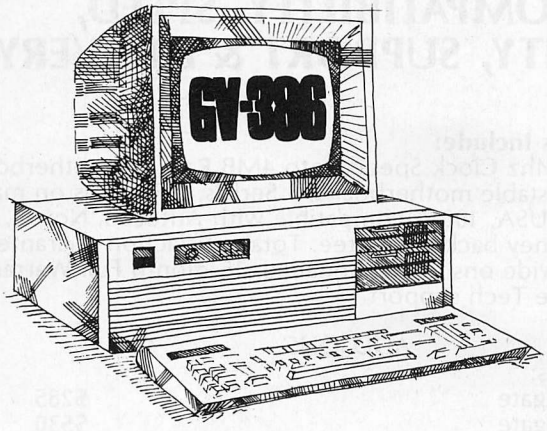
And every GV-386 is bundled with Desqview 1.3 from Quarterdeck Office Systems, giving you both expanded memory management and multitasking capabilities allowing up to nine simultaneous operations.

Affordable power

If all this speed and performance doesn't blow you away, take a look at the price: The standard system starts at around $4000. And it's available now.

The fact is, it's the most innovative compatible to date. So why settle for an 80386 machine that just "advances" technology, when the GV-386 blows it away?

And those that blow it away.



PC Designs

2500 N. Hemlock Circle
Broken Arrow, OK 74012

19 Rector St., Suite 2705
New York, NY 10006

Call us now in New York: 1-212-514-7280 or in Tulsa: 1-918-251-5550

CIRCLE NO. 159 ON READER SERVICE CARD

GET 1 YEAR ON SITE SERVICE FREE

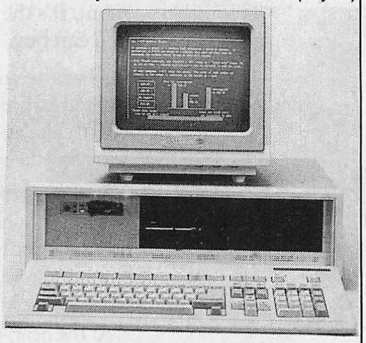
B.E.S.T. 286 AT™ Compatible

640 KB RAM Bare Bone System: only **$999** (Qty. 2)

★ FREE ★
- ☐ DOS 3.1 ($85 Value)
- ☐ 1 Year Nationwide on site service (Parts & Labor)

STANDARD FEATURES:
- Clock/calendar w/battery back-up
- *FCC and U.L. APPROVED*
- 6/8 MHz 80286 Microprocessor
- 80287 Math Co-processor Socket
- 8 Expansion Slots
- 640 KB RAM Expandable to 1024 KB on Mother Board
- 200 WATT U.L. Approved Power Supply
- AT case with Lock and LED indicators
- 5060 Compatible Keyboard
- MS DOS 3.1 ($85 value FREE)
- Full Documentation
- Complete technical support
- Fully compatible with IBM-AT

30 day money back guarantee



Evaluation Unit $1,095
We Welcome VAR and Dealer Inquiry
(quantity discounts available)

All major credit cards accepted

SYSTEM CONFIGURATION B:
- AT 640 KB RAM Bare Bone System
- 1.2 MB Teac Floppy Disk Drive
- 30 MB Hard Disk (39MS)
- Western Digital WA-2 Floppy & Hard Disk Controller
- 1 Year Nationwide (50 States) ON-SITE Service (Includes Parts & Labor)
- MS DOS 3.1

only **$1,995**

GET OUT OF THE LOOP!

NO MORE RETURNS OR LONG WAITS FOR REPAIRS
WE WILL FIX YOUR COMPUTER ON SITE
FREE OF CHARGE FOR 1 YEAR

BUSINESS ENGINEERING SCIENTIFIC TECHNOLOGIES
1914 W. Farwell • Chicago, IL 60626
(312) 465-8886 or (312) 262-3480

Prices subject to change without notice
IBM AT is a trademark of International Business Machines Corporation

Turbo 286 - AT Compatible $1295

- 8 expansion slots
- 200 Watt Power Supply
- 640 K RAM
- 6/8 MHZ Switchable from keyboard
- 2 Floppy & Hard Disk Controller
- 1.2 MB Floppy Disk Drive
- Serial, Parallel Ports
- Color Graphics Card
- AT keyboard

MEDICAL & DENTAL
- Billing & Posting
- Faster Collection
- Improved Cash Flow
- Easy Insurance Procedure
- Appointment Scheduling
- Management Reports

LEGAL SOFTWARE
- Real-Time, Multi-User
- Time & Billing
- Client/Matter Reports
- Aged A/R Report
- Matter Account Inquire
- Transaction Entry & Editing

CONSTRUCTION
- Fully Integrated Construction Accounting System
- Job Costing
- Accounts Payable
- General Ledger
- Payroll
- Accounts Receivable
- And much much more

VIDEO STORE MANAGEMENT
- Full Rental Tracking
- Membership Control
- Employee Code System
- High/Low Rental Report
- Overdue Tape Report

POINT OF SALE
- For all Retail & Wholesale Businesses
- Cash Drawer & Bar Code Scanner Interface
- Complete Audit Trail of all P.O.S. Transactions
- Inventory Inquiry during sale, and much more. . .

---

proteus
Technology Corp.

NO OTHER CLONE MATCHES PROTEUS IN IBM COMPATIBILITY, SPEED, RELIABILITY, SUPPORT & DELIVERY



Editor's Choice: "Proteus 286: the great performer" Editor, Business Computer Digest

Proteus Features Include:
- 12.5 Mhz Clock Speed upto 4MB Ram on motherboard.
- 8-layer stable motherboard, 3 Serials, 2 Parallels on mainbd
- Made in USA, 100% compatible with Autocad, Novell, Zenix
- 30-day money back guarantee. Total satisfaction guaranteed
- Free nationwide onsite maintenance, 15-month Full Warranty
- 24-hour online Tech support

PROTEUS-286GT includes:
80286-10 Intel CPU running at 12.5
1MB RAM expandable to 4MB on mainbd
8-layer stable motherboard
8 I/O slots, keybd select speed
3 serials, 2 Parallels on mainbd
Clock/cal, battery backup
Hard Disk & Floppy Controller
5 Mhz DMA bus
200 Watts quality Power Supply
Two Floppy Drives (1, 1.2MB, 1, 360K)
High Quality AT keyboards
Well Written Manuals & Utilities
12.6/6Mhz system price $2495
10/6Mhz system price $2195
10/6Mhz zero wait system $2395
12.5Mhz, 40Megabyte EGA system $3695

ADD-ONS

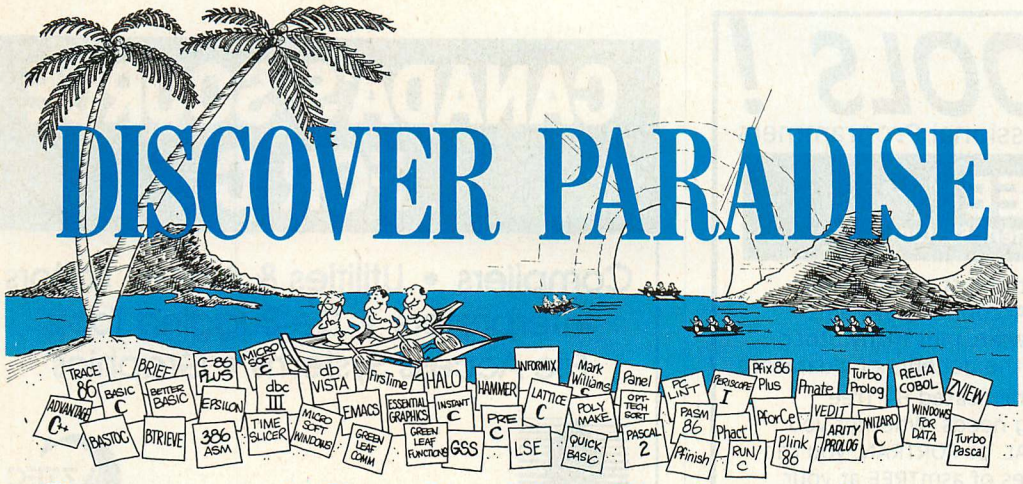
Hard Disks:	
20MB Seagate	$285
30MB Seagate	$530
40MB Seagate	$695
80MB Seagate	$1095
Monitors:	
Monochrome	$109
Color monitor	$295
NEC EGA monit.	$585

Various EGA, CGA, & Monochrome cards available. call.

To order or for information: Call
PROTEUS TECHNOLOGY CORP.
377 Route 17, Airport 17 Center
Hasbrouck Heights, New Jersey
07604

201-288-8269
Telex 510 601 0960
Fax 288 1059

DISCOVER PARADISE



Programmer's Paradise Gives You Superb Selection, Personal Service and Unbeatable Prices!

Welcome to Paradise. The MS/PC-DOS and XENIX software source that caters to your programming needs.

Discover the Many Advantages of Paradise...

- Lowest price guaranteed
- Latest versions
- Huge inventory, immediate shipment
- Knowledgeable sales staff
- Special orders
- 30-day money-back guarantee

We'll Match Any Nationally Advertised Price.

	LIST	OURS
C++		
ADVANTAGE C++	$ 495	469
PFORCE++	395	CALL
C COMPILERS		
C-86 PLUS	497	445
DATALIGHT – C	60	49
DATALIGHT – C DEVELOPER'S KIT	99	79
LATTICE C 3.2	500	269
LATTIC C W/SOURCE	900	545
LET'S C	75	59
W/CSD DEBUGGER	150	109
MICROSOFT C 4.0	450	275
MARK WILLIAMS C	495	289
SUPERSOFT C	395	339
WIZARD C	450	369
C INTERPRETERS		
C-TERP	300	235
INSTANT C	500	379
INTRODUCING C	125	105
RUN/C	150	89
RUN/C PROFESSIONAL 1.1	250	169
ASSEMBLERS, LINKERS		
386ASM/LINK	495	445
ADVANTAGE LINK	395	349
MACRO-86	150	98
PASM-86	195	125
PLINK 86 PLUS	495	325
QUELO 68000 X-ASM	595	509

Lattice Specials

	LIST	OURS
C COMPILER	500	269
C CROSS REFERENCE GENERATOR	50	35
C FOOD SMORGASBORD	150	95
C-SPRITE	175	125
CURSES	125	85
DBC-III	250	175
LMK	195	135
RPG II COMPILER	750	635
RPG COMBINATION PACK	1100	939
SECRET DISK	120	85
SIDETALK	120	85
TEXT MANAGEMENT UTILITIES	120	85

	LIST	OURS
GRAPHICS		
ADVANTAGE GRAPHICS	295	225
ESSENTIAL GRAPHICS	250	195
GRAPHIC	350	289
GSS GRAPHICS DEVELOPMENT TOOLKIT	495	379
GSS KERNEL SYSTEM	495	379
GSS METAFILE INTERPRETER	295	239
GSS PLOTTING SYSTEM	495	379
HALO—ONE LANGUAGE	300	209
W/TEN FONT PACK	425	297
HALO—FIVE MICROSOFT LANGUAGES	595	415
METAWINDOWS	185	115
METAWINDOWS PLUS	235	189
METAFONTS	80	59
METAFONTS PLUS	235	189
C UTILITY LIBRARIES		
ASYNC MANAGER	175	135
BASIC C	175	129
C ESSENTIALS	100	85
C FOOD SMORGASBORD	150	95
W/SOURCE	300	188
C TOOLS PLUS	175	135
ESSENTIAL C UTILITY LIBRARY	185	135
ESSENTIAL COMMUNICATIONS	185	135
W/BREAKOUT DEBUGGER	250	195
GREENLEAF FUNCTIONS	185	135
GREENLEAF COMM	185	135
THE HAMMER	195	175
MULTI C	149	135
MULTI COMM	149	135
PFORCE	395	245
TIMESLICER	295	265
W/LIBRARY SOURCE	1000	CALL
TOPVIEW TOOLBASKET	250	189
SCREEN DISPLAY, WINDOWS FOR C		
C WORTHY	295	269
CURSES	125	85
W/SOURCE	250	184
GREENLEAF DATA WINDOWS	225	179
W/SOURCE	395	339
FLASH UP WINDOWS	75	68
MICROSOFT WINDOWS DEVELOPMENT SYSTEM	500	319
ON-LINE HELP	149	109
PANEL	295	219
POLYWINDOWS	CALL	CALL
SCREENPLAY (LATTICE)	150	135
SOFTSCREEN HELP	195	175
VIEW MANAGER	275	199
VITAMIN C 3.0	225	199
VC SCREEN	99	84
WINDOWS FOR C	195	145
WINDOWS FOR DATA	295	239
Z VIEW	245	189
FILE MANAGEMENT		
BTRIEVE	245	195
XTRIEVE	245	195
W/REPORT GENERATION	390	315
BTRIEVE/N	595	465
XTRIEVE/N	595	465
W/REPORT GENERATION	940	750
C TREE	395	329
R TREE	295	249
C TREE/R TREE BUNDLE	650	529
CQL	395	329
DBC III	250	175
W/SOURCE	500	379
DBC III PLUS	750	599
DB VISTA	195	155
W/SOURCE	495	425
DB QUERY	195	155
W/SOURCE	495	425
FABS	150	129
FABS PLUS	195	169
INFORMIX	795	639
INFORMIX 4GL	995	799
INFORMIX SQL	795	639
PHACT	295	265

	LIST	OURS
MAKE, LINT, PROFILE, UTILITIES		
C CROSS REFERENCE GENERATOR	50	35
LMK	195	135
POLYMAKE	99	78
OTHER POLYTRON	CALL	CALL
PMAKER	125	89
PFINISH	395	235
THE PROFILER	125	94
PC LINT	139	105
PRE-C	295	159
TEXT MANAGEMENT UTILITIES	120	85
DEBUGGERS		
ADVANCED TRACE 86	175	129
BREAKOUT	125	99
CODESMITH 86	145	105
C SPRITE	175	125
CSD SOURCE DEBUGGER	75	59
PERISCOPE I 3.0	345	293
PERISCOPE II 3.0	175	145
PERISCOPE II-X 3.0	145	109
PFIX 86 PLUS	395	235
XVIEW 86	60	49

March BUNDLE of the Month

RUN/C Pro- Best-selling C interpreter PLUS Greenleaf Functions or C Utility Library. Convenience disk included—One command loads library!

LIST TOGETHER $435 **OURS** $289

	LIST	OURS
EDITORS		
BRIEF	195	CALL
CVUE	75	59
W/SOURCE	250	195
EDIX	195	155
EMACS	295	265
EPSILON	195	159
FIRSTIME (C)	295	229
KEDIT	125	105
LSE	125	89
PMATE	195	119
PC/VI	149	129
SPF/PC	195	149
VEDIT	150	109
VEDIT PLUS	185	139
ADDITIONAL PRODUCTS		
DAN BRICKLIN'S DEMO PROGRAM	75	59
FASTBACK	175	149
INTERACTIVE EASYFLOW	150	129
PDISK	195	129
SOURCE PRINT	97	87
TREE DIAGRAMMER	77	69
VENTURA PUBLISHER (XEROX)	895	805
PASCAL COMPILERS		
MICROSOFT PASCAL	300	189
PASCAL 2	350	329
TURBO PASCAL	100	69
OTHER BORLAND	CALL	CALL

	LIST	OURS
TOOLS FOR TURBO PASCAL		
ALICE	95	68
FIRSTIME	75	59
FLASH UP WINDOWS	90	79
TURBO HALO	129	99
SCREENPLAY	100	89
SCREEN SCULPTOR	125	94
T-DEBUG PLUS	60	50
TURBO EXTENDER	85	65
TURBO PASCAL ASYNC MGR	100	84
TURBO PROFESSIONAL	70	49
TURBO POWER TOOLS PLUS	100	83
TURBO WINDOWS	80	65
OTHER TURBO TOOLS	CALL	CALL

NEW Products

ADVANTAGE C++ for XENIX—Take advantage of object-oriented programming methods. Add resiliency and flexibility to your code. Build large and sophisticated programs more productively.
List $695 Ours **$660**

ADVANTAGE Make—Feature-packed MS/PC-DOS version of UNIX MAKE utility.
List $125 Ours $99

SSP/PC—Fast, extremely accurate library of over 145 math subroutines. Callable from C, FORTRAN, Pascal, BASIC.
List $350 Ours **CALL**

TIMESLICER—New Microsoft version. Multitasking, linkable library supporting concurrent tasks and real-time event processing with header files provided for C++, C and assembly. Library source available!
List $295 Ours $265

VENTURA PUBLISHER (XEROX)—Desktop publishing software, lightning fast, loaded with features. Create professional-looking documentation at minimal cost!
List $895 Ours $805

	LIST	OURS
BASIC		
BETTERBASIC	199	139
SUMMIT ADD ONS	CALL	CALL
BETTER TOOLS	95	89
FINALLY	99	89
MICROSOFT QUICKBASIC	99	75
PROFESSIONAL BASIC	99	75
8087 MATH SUPPORT	50	45
PANEL-BASIC	145	115
TRUE BASIC	150	105
ADD ONS	CALL	CALL
COBOL COMPILERS/UTILITIES		
MICROSOFT COBOL	700	445
MICROSOFT COBOL TOOLS	350	205
MICROSOFT SORT	195	139
MICRO/SPF	175	CALL
OPT-TECH SORT	149	115
REALIA COBOL	995	785
SCREENPLAY	175	155
RM/COBOL	950	639
RM/COBOL 8X	1250	895
VISUAL COBOL (MBP)	1150	1015
FORTRAN COMPILERS/UTILITIES		
LAHEY FORTRAN	477	CALL
MICROSOFT FORTRAN	350	209
RM/FORTRAN	595	389
ACS TIMES SERIES	495	419
87 SFL	250	225
FOR-WINDS	90	78
FORLIB-PLUS	70	54
GRAFMATICS OR PLOTMATICS	135	119
GRAFMATICS AND PLOTMATICS	240	219
FORTRAN SCIENTIFIC SUBROUTINES	295	249
STRINGS AND THINGS	70	54

XENIX/UNIX SOFTWARE

	LIST	OURS
XENIX SYSTEM V (COMPLETE SYSTEM)—SCO	1295	995
SYSTEM V/AT—MICROPORT	440	395
OTHER SCO AND MICROPORT	CALL	CALL
ADVANTAGE C++	695	660
BTRIEVE	595	465
C-ISAM	319	285
C TREE	395	329
MICROSOFT BASIC	350	239
MICROSOFT COBOL	995	635
MICROSOFT COBOL TOOLS	450	205
MICROSOFT FORTRAN	695	439
MICROSOFT PASCAL	695	439
PANEL	625	545
RM/COBOL	1250	949
RM/FORTRAN	750	549

	LIST	OURS
ADDITIONAL LANGUAGES		
APL PLUS	595	429
JANUS ADA/C PACK	95	89
LOGITECH MODULA 2	89	63
PC/FORTH	150	119
SMALLTALK V	99	88
TURBO PROLOG	100	75

CALL FOR OTHERS/ADD-ONS!

Terms and Policies

- **We honor MC, VISA, AMERICAN EXPRESS**

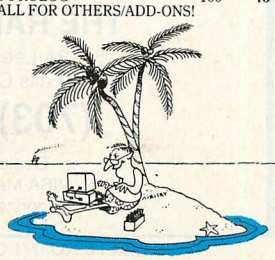
No surcharge on credit card or C.O.D. Prepayment by check. New York State residents add applicable sales tax. Shipping and handling $3.00 per item, sent UPS ground. Rush service available, prevailing rates.

- Programmer's Paradise will match any current nationally advertised price for the products listed in this ad.
- **Mention this ad when ordering**—some items are specially priced.
- Prices and Policies subject to change without notice.
- Corporate and Dealer inquiries welcome.

1-800-445-7899 In NY: 914-332-4548

Programmer's Paradise
42 River Street, Tarrytown, NY 10591
914-332-4548

Programmer's Paradise



CIRCLE NO. 143 ON READER SERVICE CARD



TOOLS !

for Professional Programmers

asmTREE™

Database Development System

A database system is the heart of just about every successfull software package ! Accounting systems, Mailing programs, Analysis software, plus many other types of packages have reaped the benefits of a database.

NOW you too can utilize a sophisticated B+Tree database for your programming needs. Whether you program in Assembler, C, PASCAL, or FORTRAN, you can have all the advanced capabilities of asmTREE at your finger tips. asmTREE is written in assembly language for fast execution and small code size - for the DOS 2.xx or newer environment.

ISAM Functions - NO ROYALTIES - Full SOURCE CODE

asmTREE™ - The Programmer's Database

Only $395.00 - Complete

Other fine development tools -

- ASMLIB - The Programmer's Library - Over 210 functions for Assembler, C, PASCAL, and FORTRAN. NO ROYALTIES. With SOURCE CODE $149
- FPLIB - IEEE FLoating Point for REALIA™ COBOL $149
- Turbo.ASM - Assembly Langauge interfacing made easy for TURBO PASCAL ... $99.95

CALL TOLL FREE 1-800-262-8010
in Calif. CALL 1-714-526-5151

BC Associates
3261 N. Harbor Blvd - Suite B
Fullerton, CA 92635

VISA

CANADA'S SOURCE FOR C

Compilers • Utilities & Aids • Editors
Interpreters • De-Bugging Tools
File Access Systems • Graphics

GREENLEAF SOFTWARE©

AZTEC

Phoenix

Lattice

LIFEBOAT™

Complete Line of Programming Development Tools
Full Service and Support - Fast Delivery

VISA CORPORATE DISCOUNTS MasterCard

(416) 449-9252/5

SCANTEL SYSTEMS LTD.

801 YORK MILLS RD., 201, DON MILLS, ONT M3B 1X7

MEGAMEMORY AND DESKTOP PUBLISHING

Lowest Prices In USA

Fully Populated 2MB Boards
Made by Tall Tree Systems
HIGHEST QUALITY RAM CHIPS

JRAM-2	$319
JRAM-3 LOTUS-INTEL	$389
JRAM-AT	$389
JRAM-AT3 LOTUS-INTEL	$429
JLaser-Plus PC	$599

SUPER SPECIAL

OMS KISS Laser Printer
W/TWO MEGABYTE JRAM-3
and JLASER-PLUS $2499
600x300 Dots Per Inch!

THE RAM EXPLOSION
5119A Leesburg Pike, Suite 260
Falls Church, VA 22041

(703) 569-4471

Dealer Inquiries Invited
VISA/MASTERCARD AMEX
An Authorized TALL TREE DEALER

CIRCLE NO. 171 ON READER SERVICE CARD

EVSAN COMPANY (415) 991-1051
P.O.BOX 2143 DALY CITY, CA 94017

DYNAMIC RAMS

1 MEG	100ns	65.00
41256	100ns	4.75
41256	150ns	2.75
41256	120ns	3.10
4164	120ns	1.20
4164	150ns	1.00
4464	120ns	4.25
4464	150ns	3.95
4116	150ns	.98

SPEED UP YOUR IBM PC 20-30%
REPLACE 8088/8086 with:

V-20	8mhz	9.95
V-20	5mhz	8.95
V-30	8mhz	11.00

EPROMS

27512	250ns	18.75
27256	250ns	5.50
27C256	200ns	7.10
27C256	250ns	6.75
27128	150ns	5.75
27128	250ns	3.60
2764	200ns	3.75
2764	250ns	3.20
27C64	200ns	4.90
2732A	200ns	3.90
2732A	250ns	3.50
2564	450ns	7.50
2532	450ns	4.10
2716-1	350ns	3.70
2716	450ns	2.95
2708	450ns	2.50

8000's

8031AH	5.25	8243	2.00
8035	1.95	8250A	3.95
80C35	3.75	8251A	1.65
8039	2.50	8253-5	1.75
80C39	4.00	8254	3.50
8085A	1.75	8255A-5	1.80
80C85	3.75	8272	3.50
8086	5.00	8274	4.75
8155	1.60	8284	2.95
8156	2.25	8288	5.25
8212	1.60	8748H	6.25
8216	1.50	8749H	8.25
8226	1.75		
8237A-5	4.75		

INTERFACE

1488	.32	1489	.32

PRIME PARTS
100% GUARANTEED

MATH CO-PROCESSORS

C8087-2	8mhz	142.00
C80287-6	6mhz	168.00
C80287-8	8mhz	275.00

STATIC RAMS

43256L	120ns	25.50
6264L	100ns	3.95
6264L	120ns	3.60
6264L	150ns	2.85
6264P	150ns	2.65
6116P	150ns	1.45
2015B	100ns	1.75
4016	150ns	1.60
4016	200ns	1.30
2114A	120ns	1.50
6147	35ns	3.25

COLOR GRAPHIC CONTROLLER:

D7220AD	18.50

MOTHERBOARDS

XT Motherboard	$ 149.00
XT TURBO BOARD	210.00
AT Motherboard	999.00

IBM COMPATIBLE INTERFACE CARDS

Floppy Disk Drive Adaptor	$ 45.00
Color Graphic Adaptor	80.00
Monographic Card	99.00
Multifunction Cards	95.00

FLOPPY DISK DRIVES

TEAC 5¼" FD55B	94.00
FUJITSU 5¼" M2551	82.00

DIGITAL REAL TIME CLOCK

5832	2.80
6242	4.90

Z80 FAMILY

Z80A CPU	4mhz	1.20
Z80A CTC	4mhz	1.20
Z80A PIO	4mhz	1.20
Z80A DART	4mhz	2.75

STATIC RAMS

5564PL	150ns	5.25
5565PL	150ns	3.25

PROMPT DELIVERY

TERMS & CONDITIONS:
1) Visa & Mastercards Accepted with 3% surcharge.
2) Prices subject to change. PLease call for current & volume pricing.
3) Shipping & Handling (1 lb)
UPS Surface $3.00
UPS 2nd Day $4.50
California Residents add 6.5% sales tax.

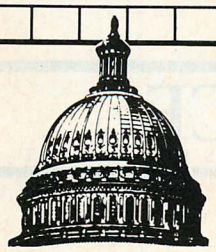
OFFICE HOURS:
Monday thru Friday 7:30AM - 5:30PM
Saturday 7:30AM -12:00Noon

Data Sheets: $0.25 each

Quarterly Flyers available, please call

We reserve the right to substitute manufacturer. All merchandise subject to prior sale.

CIRCLE NO. 112 ON READER SERVICE CARD



"Serving the Nation's Capitol and the World"

ITS TEK-NET-BBS (703) 690-7462

Visa, MC, CHOICE, AE
Leasing, Renting & Financing available

Open 7 days a week

(703) 847-4740 (800) 642-2395

Information and Technology Services, Inc.

Micro Systems Specialists

8478A Tyco Rd., Vienna, VA 22180

Drives

20 MB SEAGATE 65MS	$379
30 MB SEAGATE RLL	$479
20 MB ST4026 AT	$573
30 MB ST4038 AT	$651
40 MB ST4051 AT	$792
80 MB ST4096 AT	$1273
40 MB PRIAM XT	$1395
60 MB PRIAM XT	$1450
40 MB PRIAM AT	$1175
60 MB PRIAM AT	$1350
BERNOULLI DUAL 10	$1839
BERNOULLI DUAL 20	$2494
20MB PLUS HARDCARD	$695
20MB MAYNARD HCARD	$759
ISI WORM 220MB INT.	$2995
60MB ARCHIVE TAPE	$740
60MB ARCH. EXT. TAPE	$740
60MB GENOA TAPE	$935
60MB SYSGEN	$1089
TOSHIBA 3.5 DRIVE	$150
*TOSHIBA 5.25 EXT.	$349
FUJITSU 360 K DRIVE	$89

Laser Printers

CANON A1	$2095
CANON A2	$3085
HP LASERJET	$2295
QMS KISS	$1995
XEROX 4045	$CALL

Printers

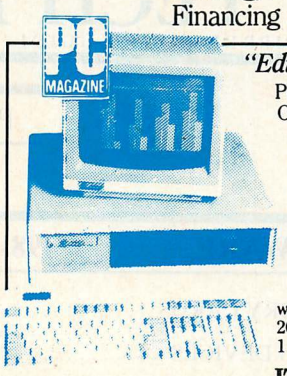
NEC P-6	$465
NEC P-7	$670
NEC P-5	$1107
NEC P-5XL	$1230
PANASONIC 1080 I	$220
PANASONIC 1091 I	$284
PANASONIC 1092	$345
PANASONIC 1592	$465
STAR LV 1210	$174
STAR NX-15	$368
STAR ND-15	$430
STAR NR-15	$522
STAR NB 24-15	$684
STAR SD-10	$340
STAR NX-10	$240
BROTHER 1509	$425
CITIZEN MSP-10	$305
CITIZEN MSP-15	$420
CITIZEN PREMIER 35	$470
TOSHIBA P341	$820
TOSHIBA P351	$1212
FUJITSU DLP24	$1224
FUJITSU DM91	$398
OKIDATA M182	$264
OKIDATA M192+	$397
OKIDATA M193+	$598
OKIDATA M84	$698
OKIDATA 2410	$1961
EPSON FX-286	$609
EPSON LX-80	$270
EPSON FX-85	$419
EPSON LQ-800	$616

Memory Boards

JRAM 2	$129
JRAM 3 ABOVEBOARD	$179
JRAM AT	$179
JRAM AT3 ABOVEBOARD	$239
JLASER MODULE	$265
JLASER+	AVAIL.
INTEL ABOVEBOARD	$438
AST RAMPAGE 512K	$485
ORCHID CRAMRAM	$291
ZUCKER BOARD	$68
AST 6 PAK+W/384	$210
QUADBOARD W/384	$199

Chips

64K 150ns set of 9	$8
64K 120ns set of 9	$9
256K/150ns set of 9	$23
256K/120ns set of 9	$25
64 × 4	$4
128K Piggyback	$5
8087-3	$119
8087-8	$169
80287 6MHz	$192
80287 8MHz	$306



"Editor's Choice"
PC MAGAZINE
OCT. 14, 1986
The ITS Turbo XT from Information and Technology Services is our "best buy" recommendation

$849.00 complete
with 20MB and 1 floppy **$1157**

ITS Turbo XT
1 year warranty
30 day money back guarantee

IBM® PC Compatible

ITS TURBO 286

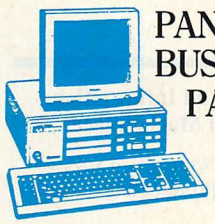
$1195.00



- Phoenix ROM BIOS
- Full one year warranty
- 80286 CPU switch selectable 6 or 8 Mhz (10 Mhz optional)
- 1.2 MB Floppy Disk Drive
- Supports up to 5 internal ½ HT drives
- 2 serial/1 parallel ports built into chassis
- real time clock • 8 expansion slots

and many more features

- 6/10 Mhz optional $1295



PANASONIC BUSINESS PARTNER

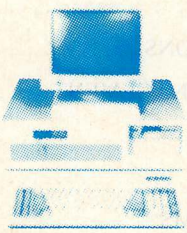
FX-600 2DS/DD, 640K
$895
with 1 DS/DD, 20 MB
$1145
(Monitor and Adapter extra)

PRICING SENSATIONS!

20 MB SEAGATE DRIVE	$379
30 MB SEAGATE DRIVE	$479
SAMSUNG AMBER MONITOR	$79
NEW ENHANCED KEYBOARD	$79
MONO GRAPHICS CARD	$90

OTHER SYSTEMS

BIOS AT 8Mhz	$1769
SPERRY MICRO IT	$1795
IBM XT	$1195
IBM AT	$2395
1800+ AT (EVEREX)	$1195



ITS SYSTEM 386 *BLACKHAWK*

Features:

- INTEL 80386 CPU and support circuits
- Phoenix BIOS
- 18 Mhz clock speed
- PC/AT compatible 8 Mhz switchable from keyboard
- 512K RAM standard up to 14 megabytes
- Parallel/Serial/Clock
- 8 Slot Expansion bus interface
  2-PC Compatible 8 bit bus connectors
  2-32 bit bus connectors
- Hard disk/Floppy disk controller
- 1.2 megabyte floppy

Available now
$3995.00

LAPTOPS and ACCESSORIES

Toshiba T1100 w/640	$1745
Toshiba 3800	$3395
Sharp PC 7000	$995
NEC Multispeed	CALL
Zenith 181	CALL
Panasonic Exec. Partner	CALL
Centronics GLP II Printer	$215
Toshiba Ext. Drive	$349
Toshiba 1200 Modem	$336

NOVELL NETWORKING DESIGN & INSTALLATION

- Prices subject to change 1/16/87
- 10% re-stock fee on all items
- Software non-returnable if opened
- No surcharge on VISA, MC, CHOICE...AE 3%



SPERRY

PC/IT

Superior to the IBM AT
Rated 8.8 by InfoWorld

44MB, 28ms Access Hard Drive
3 Speed Processor
Free 8 MHz Math Coprocessor
1.2MB Floppy, 640K Ram
2 Serial/1 Parallel Ports
Clock/Calendar, AT Keyboard
DOS 3.1 Basic, System Guide

$2,695.00 1 year warranty

CIRCLE NO. 134 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PRODUCT CATEGORIES

HARDWARE 198-199

ACCESSORY CARDS 197
COOLING DEVICES .....
COMMUNICATIONS .....
COMPATIBLES .....
GENERAL 198
MASS STORAGE .....
PERIPHERALS 198-199
SECURITY DEVICES .....
USED EQUIPMENT .....

SOFTWARE 199-205

ACCESSORIES/SUPPLIES 199
ARTIFICIAL INTELLIGENCE 199
BUSINESS 199
COMMUNICATIONS 199
DATA BASE MANAGEMENT .....
EDUCATIONAL .....
ENGINEERING 199
EXPERT SYSTEMS 199
GENERAL 199-200
GRAPHICS 200
LANGUAGES 200
MULTI/USER SYSTEMS .....
NETWORKING .....

SOFTWARE *continued*

OPERATING SYSTEMS 200
PROGRAMMERS TOOLS 200-202
PUBLIC DOMAIN 202
SCIENTIFIC 202-203
SECURITY DEVICES 203
STATISTICS 203
TAXES 203
TERMINAL EMULATION 203
UTILITIES 203-205
WORD PROCESSING .....

MISCELLANEOUS 205

BAR CODING 205
PUBLICATIONS 205

Advertising Rates and Information:
PC Tech Journal Marketplace

PC Tech Journal Marketplace is a special economical section for product and service listings.

Listings are grouped by category and sold by column inches. Second color option available.

Standard Directory Listings are also available for a minimum of 3 issues at $170 per issue ($510 total).

For additional information
call 212-503-5115.

PC Tech Journal Classified Advertising Staff
One Park Avenue, New York, NY 10016
(212) 503-5115

Advertising Director
Kathryn J. Cumberlander

Sales Manager
Daniel L. Rosensweig

Sr. Advertising Coordinator
Monica Dixon

Advertising Coordinator
Angela Kiffin

Sales Assistant
Linda Annis

Production Manager
Anne R. Brockinton
(212) 503-5441

Production Coordinator
Elliot Appel
(212) 503-5470

Account Managers

Lisa B. Stick (212) 503-5172

AL, AR, IA, IL, IN, KS, KY, AZ, CO, OR, NM, LA, MI, MN, MO, MS, NB, ND, OH, OK, SD, TN, TX, NV, AK, GA, UT, CA (ZIP 92999 & DOWN), CANADA (OTHER THAN BRITISH COL.) AND ALL OVERSEAS CALLS.

CT, MA, ME, NH, NJ, NY, RI, ID, MT, MD, VT, DC, DE, HI, NC, SC, FL, VA, WV, WI, PA, WA, WY, CA (ZIP 93000 & UP) BRITISH COL.

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

A STEP BEYOND

IN 80386 TECHNOLOGY

SWITCH YOUR SLOW IBM "AT/XT286" INTO A FAST 386!!

WITH THE KW386-ET16

FEATURES:

- 75 TO 250 % FASTER THAN "COMPAQ"
- 80386 SYSTEM SPEEDS OF 12 TO 24 MHZ AND AT/XT286 BUS SPEED OF 6 TO 12 MHZ
- SELECTABLE 80287/80387 MATH CHIP SPEEDS AT 8, 10, 12, 14, 16 MHZ
- HIGH SPEED MEMORY EXPANDABLE TO 16 MEG ON BOARD
- BGI 386 BUS INTERFACE EXPANSION CONNECTOR ON REAR OF THE BOARD WILL PROVIDE COMPATIBILITY WITH "IBM 386/RISC COMPUTER SYSTEMS
- KW CPU BOARD WILL TAKE ONE 16 BIT EXPANSION SLOT
- BUILT IN 386 BIOS WILL INTERFACE WITH "IBM BIOS" TO PROVIDE 100 % SOFTWARE & HARDWARE COMPATIBILITY

BGI COMPUTER DIV. (215)538-3900

CIRCLE 375 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

HARDWARE

ACCESSORY CARDS

Z80 and HD64180 CO-processors
For PC, PC/AT. Clock speeds to 9mhz. Prices start at $199.50. Run CP/M-80 software fast. Develop code for Z80/HD64180 with software ICE. Run Intel ISIS tools. Interface to real world with iSBX bus devices. High speed communications, including Apple Talk compatible.
Decmation
2065 Martin Ave. #110
Santa Clara, CA 95050
(408) 980-1678

PC ANALYZER
Real-Time debugging package for your PC or XT. Complete with board and debugging software. Also allows you to use your own software debugger. Nonintrusive operation, simple to install. Operates with DOS & QNX. Price $995. Free shipping.
Sofpak Technologies, Inc.
215 Stafford Road, Unit 101
Ottawa, Canada K2H 9C1
(613) 726-1908

DIGITAL SIGNAL PROCESSOR
The Model 10 coprocessor board is based on the 16/32 bit TI TMS 32010 and is designed for applications in communications, speech, instrumentation, and numeric processing. A 1K complex FFT takes 90ms. Offered with onboard 12 bit 40 Khz A/D and D/A. Includes all utility and applications software. $650-$850.
Dalanco Spry
Suite 241 2900 Connecticut Ave. NW
Washington, DC 20008
(202) 232-7999

PC-PROMPAK
ROM Expansion for PC!
Aldia systems introduces PC-PROMPAK, a "half-sized" PROM/ROM expansion board for IBM and IBM compatible PCs. PC-PROMPAK will support up to six 28-pin JEDEC compatible devices (ex: 2764, 27128, 27256, 27512, 6264, etc.) with individually selectable address ranges. Prices start at $125 for single units. Quantity discounts and OEM arrangements available. MC/VISA.
ALDIA SYSTEMS, Inc.
P.O. Box 37634
Phoenix, AZ 85069
(602) 866-1786

FIXED DISK BIOS/BOOT
FiXT boots from most popular Hard Disks—DAVONG, TECMAR, IOMEGA, GT LAKES, etc. Adds XT-like BIOS interface to your disk for PC. Security, multiple volumes, removable media support optional. No-slot plug-in installation. Specify controller and computer with order. $80-$95. Add $3 shpg., CA tax.



GOLDEN BOW SYSTEMS
2870 Fifth Avenue
Suite 201
San Diego, CA 92103
(619) 298-9349

Truly Low Cost PC Imaging!



IMAGE ACE II
Video Capture System
- Digitize video from cameras, tuners, and VCRs directly to your IBM PC display
- 320 x 200 x 4 levels
- 1.3 sec. full screen capture
- Complete with hardware card, software, cable, and manual

LODGE ELECTRONICS
P.O. Box 338 • Streamwood, IL 60103
(312) 837-6553
CIRCLE 378 ON READER SERVICE CARD

PC-SPRINT

"PC-Sprint is the most cost effective PC Speedup product on the market."
—Computer Shopper Magazine

- Run your PC, XT or clone at 7.38 mhz.
- 280% Speedup (Norton SI rating)
- Speeds up all software—you can see the difference
- External speed switch
- External reset button
- Change speed "on the fly"
- Compatible with 8087
- Works with all color or mono displays
- "Slotless" plug-in on most PCs
- Includes: Selectable top speed, instructions, warranty, tool, remote mount switch, free BBS subscription

$89.95 V20 add $10. Call for information on other products

Exec-PC, Inc.
P.O. Box 11268 Shorewood, WI 53211
(414) 242-2173
CIRCLE 377 ON READER SERVICE CARD

IBM-AT SPEEDUP

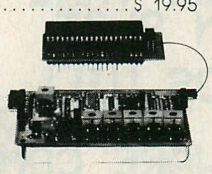
The High-Performance SpeedInjector™ from Ariel

As the industry's recognized leader in high-performance speedup products, Ariel has performed extensive research and developed unmatched experience in this field. Our products offer the complete solution.

- **XCELX 286/287 XPRESS™** A SpeedInjector for ALL IBM-ATs. Uses reliable frequency synthesis for full compatibility and high performance • 100% variable from 5-13 MHz CPU speed, while running • Mode switch defaults to standard 6 MHz or fast mode • Hardware reset switch • Speedup the 80287 independently. Choose from: Standard—⅓ CPU speed, 8, 10, 12, 14, or 16 MHz actual co-processor speed • Rear mounted • One-year warranty ..... $ 99.95
- **XCELX 286/287 XPRESS + The Speed Utilities**—The SpeedInjector with software that will display exact XCELX frequencies • speedup hard disk by 50% • speedup keyboard reaction time • correct floppy disk access $139.95
- **FAST 80286-10**—For CPU speeds of 10+ MHz $299.95
- **FAST 80287**—8, 10, 12, 14, 16 MHz ..... Call
- **FAST RAM**—100 & 120 NS, 128K & 256K ..... Call
- **Mil-Spec Crystals**—The famous Ariel Crystals. For early ROM ATs. Available: 16-17-18-19-20-22-24 MHz ..... $ 19.95

increase overall speed up to 300%





Ariel

ORDER HOTLINE: 201-788-9002

P.O. Box 866—Flemington, NJ 08822

CIRCLE 392 ON READER SERVICE CARD

Tech Marketplace . . . the comprehensive guide to products and services for the MS DOS market.

GENERAL

VIDEO LAN 'LINK SYSTEM'®
FOR IBM, PC, PC/XT, PC/AT labs. Instructor has complete control of all trainee computer monitors. Instructor can 1) transmit image, 2) receive trainee image or 3) transmit any trainee image to any/all trainees. Color or mono. Software independent. Increases instructor efficiency and trainee comprehension.
APPLIED COMPUTER SYSTEMS, INC.
3060 Johnstown-Utica Road
Johnstown, OH 43031
1-800-237-LINK

PERIPHERALS

SPEECH SYNTHESIS
SynPhonix: TRUE Unlimited Speech Synthesizer for IBM-PC/XT/AT/jr & compatibles. This low power short card includes an SSi263 speech chip, amplifier and speaker. Software includes Text-to-Speech, Phonetic Editor, Talking Clock & demos. Can be programmed with BASIC and other languages. Prices start below $200.

SynPhonix™
Electronic Speech Articulator

Artic Technologies
1311 N. Main St.
Clawson, MI 48017
(313) 435-4222

CP/M & 1.2Mb AT ON PC
With MULTI-DISK card & UniForm-PC use 3.5, 5.25 & 8-inch single & double density CP/M format as DOS diskettes on your IBM PC or XT. Many MS-DOS formats supported including IBM AT 1.2 Mb. HP-150 & Data General 1. Over 200 formats. Both MULTI-DISK & Uniform-PC for $225. Disk drives & adapter cable available.
PS Engineering
P.O. Box 51068
San Jose, CA 95151-5068
1-800-369-2398; 1-800-423-7171 in CA.

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PERIPHERALS

EPROM/EEPROM PROGRAMMER

Programs 2716-27512, 25xx, 68764/66 eproms via RS-282. Also 874x, micros, 28xxA & 52Bxx eeproms. Automatic Baud rate select, built in menus, no personality modules. Price: $250. Mention this ad for free terminal software. 16 BIT I/O MODULE $75
For control of input or output lines via RS-232. Use with modems for remote control.
INTELLITRONICS
P.O. Box 3263; Tustin, CA 92680
(714) 669-0614

CREATE A DISKLESS PC!

PC-ROMDRIVE allows users to create a "Diskless PC" capable of booting a ROM-resident copy of MS-DOS and/or user application programs. PC-ROMDRIVE consists of a PC-compatible ROM/PROM expansion board and the PC-ROMDRIVE software. PC-ROMDRIVE is priced at $195 for single units. Quantity discounts and OEM arrangements available. MC/VISA
ALDIA SYSTEMS, Inc.
P.O. Box 37634
Phoenix, Az. 85069
(602) 866-1786

SOFTWARE

ACCESSORIES/SUPPLIES

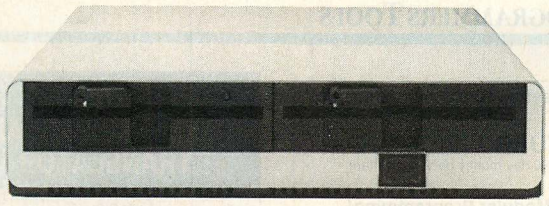
ARTIFICIAL INTELLIGENCE

TURBO EXPERT

Full Scale IBM-PC Expert Systems/Ready To Consult. $34.95. Runs on all compatibles.
#1. THE STOCK MARKET EXPERT
#2. THE EXECUTIVE HEALTH EXPERT
#3. THE PSYCHIATRY EXPERT
#4. THE TURBO EXPERT TOOLKIT III
Please specify # when ordering.
Thinking Software, Inc.
46-16 65 Place
Woodside, N.Y. 11377
(718) 429-4922

FOR INFORMATION
OR TO PLACE
YOUR AD
IN
PC Tech Journal
Marketplace
Call
(212) 503-5115

DISK
COPIER



Fast (one minute)
Simple (one button)
Reliable (one board)
$995 (one price)

RECORTEC
275 Santa Ana Ct., Sunnyvale, CA 94086
(408) 737-8441
CIRCLE 379 ON READER SERVICE CARD

BUSINESS

OPERATIONS RESEARCH

TSA88	Transportation Simplex Algorithm (up to 510 sources, sinks or transshipment points)
TNET88	Transportation Network System (networks up to 510 nodes & 16K links)
TPR088	Transportation Problem Solver (shortest path, tours up to 50 stops)

Req. 192K, color graphics adaptor. $99 each w/ 8087 support, User's guide. Write or call for our brochure.
EASTERN SOFTWARE PRODUCTS INC.
P.O. Box 15328, Alexandria, VA 22309
(703) 549-5469

Auto-Pilot™

Put your responsibilities on Auto-Pilot: ToDo list, sophisticated Tickler file, appointment calendar. Tracks employee assignments/action items. Windows display future, present, and uncompleted past events. Multi-users, multi-files, periodic & one time events. DOS 2.0+ PC/XT/AT/ compatible $29.95 Check/Visa/MC
Advanced Concepts
P.O. Box 246
Ironia, N.J. 07845
1-800 235-6646 Ext 852 1-800-235-6647(CA)

COMMUNICATIONS

PC SERIAL DATA ANALYZER

Use your IBM PC or compatible to analyze data streams between two serial devices (up to 9600 BAUD). Two windows display each devices transmission in ASCII or HEX. PC can also act as a terminal for either device. Invaluable tool for debugging serial interfaces. Disk & manual $150.
Triple C Software
2897 SW 13th St.
Fort Lauderdale, FL 33312
(305) 583-0687

ENGINEERING

ENGINEER'S AIDE

- Pipeline/Ductwork Sizing
- Pump/Fan/Compressor Sizing
- Heat Exchanger Sizing
- Orifice/Control Valve Sizing
- Project Financial Analysis
- Conversion Calculator
- Specification Writer

Pull down menus, Pop-up help windows, Single Screen entry & results-ALL above for $395 (into price, $back guarantee). For IBM & MAC.
ENGINEERING PROGRAMMING CONCEPTS
P.O. Box 925
Camarillo, CA 93011
(805) 484-5381 In CA, 1(800) 367-3585

METAL FABRICATORS

PC/Cultist takes input from your bill of material—Detail drawing and calculates the best cutting combination for any length stock and prints a shop ready cutting list and scrap report. Also an optimization feature finds best multi length for mill orders. Price $300. Demo Disk $25.00
THE JOSEPH ALBERT CO.
P.O. Box 611
Blue Island, Illinois 60406
(312) 349-9032

FINITE ELEMENT ANALYSIS

MSC, the leader in FEA technology, markets a full line of FEA tools for personal computers. Starting at $45 for MSC/pal INTRO on either the IBM PC or the Apple Macintosh, our products are designed to be complete and easy to use. Interfaces for most CAD systems available.
The MacNeal-Schwendler Corporation
815 Colorado Blvd.
Los Angeles, California 90041
(213) 259-3888

EXPERT SYSTEMS

CxPERT for Expert Systems

C programmers interested in using expert systems technology will love CxPERT. AI features such as explanations, why, frames, av pairs, legal values and more are completely compatible with C. Create executable systems with no royalties. $165 + $5 s&h. MD add 5%. CK/MO/ Visa/MC. Req. C compiler & DOS 2.0+.
Software Plus
1652 Albermarle Dr.
Crofton, MD 21114
(301) 261-0264

GENERAL

DOCUMENTATION-BY MAIL™

Technical writing service specializing in long-distance production of economical and timely manuals for small, medium-sized and large developers. Tutorials, user's guides, reference manuals. Fixed price contract, professional quality, quick turnaround. Call for credentials, sample and free estimate.
BNP Enterprises, Inc.
20370 SW 84 Ave.
Miami, FL 33189
(305) 253-2317

GREAT SOFTWARE, CHEAP!

Only $5.95 per disk for absolutely smashing Shareware and Public Domain programs! Money-back guarantee. PC-Outline, AutoMenu, PC-Key Draw, PC-DeskTeam, PC-Write 2.7, DOSamatic, Utilities Galore plus Databases, Arcade and Adventure Games. Lots more! IBM PC, PC jr, and compatibles. Send for free catalog.
SHAREWARE EXPRESS
31877 Del Obispo, Suite 102Q
San Juan Capistrano, CA 92675
(714) 240-1322

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

General

PUBLIC DOMAIN SOFTWARE IN C

Over 115 volumes of public domain software in CP/M & MS-DOS formats.
- editors & compilers
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details.

C Users' Group

THE C USERS' GROUP
P.O. Box 97
McPherson, KS 67460
(316) 241-1065

TAPE/DISK CONVERSIONS

Conversion services to or from over 500 computer systems:
- Magtapes
- Micro Computers
- Mini Computers
- Word Processors
- Typesetters

Our conversion capabilities surpass most in the industry.
Pivar Computing Services, Inc.
165 Arlington Hgts. Rd. #T
Buffalo Grove, IL 60089
(312) 459-6010

Graphics

FORTRAN GRAPHICS LIBRARY

GRAFMATIC (screen graphics): 75 MS FORTRAN/Pascal, R-M/Profort, Lahey FORTRAN callable subroutines. Fully documented, prof. graphics capabilities, inc. general utility, 2-D interactive, total 2-D plots, 3-D plots and solid models. $135. H-P or H-I plotter? get PLOTMATIC, complete plotter graphics library. Interfaces w/GRAFMATIC. $135. Both $240.
MICROCOMPATIBLES, INC.
301 Prelude Drive Dept. J
Silver Spring, MD 20901
(301) 593-0683

MetaWINDOW™/TurboWINDOW™

Advanced graphics toolkit provides Xerox Star/Apple Macintosh style graphics on your IBM PC. Supports most popular graphics cards. Allows you to create pop-up menus, windows & icons; use proportionally spaced fonts; rubberband & rag lines, text or bitmap images; supports mouse-cursor tracking. Tightly optimized for use with Turbo Pascal, IBM Pascal, C, Fortran.

METAGRAPHICS SOFTWARE CORPORATION

METAGRAPHICS SOFTWARE CORP.
4575 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-1550

CIRCLE 399 ON READER SERVICE CARD

FORTRAN TOOLS & GRAPHICS

PC-PLT: CALCOMP and VERSAPLOT Compatible Graphics Package for the Fortran Programmer. Supports CGA, EGA, Tecmar and Printer Graphics. $325
PC-TOOLS: 125 Subroutines and Functions Giving Fortran Programmers Complete access to the PC. $125
ONTAR Corporation
129 University Road
Brookline, MA 02146-4532
617-739-6607

SCIENTIFIC DATA PLOTTING

SCI-GRAF creates graphs up to 1680 X 1712 dots (over 3 million pixels!) on Epson or IBM graphics, printers. Supports log scaling, overlays, point-labeling, legend creation, batch mode, wide-carriage printers, and color graphs on a JX-80. Requires DOS 2 or 3, 256k. No credit cards. $99.95
Microcomputer Consultants (MSC)
32 W Anapamu Suite 190
Santa Barbara, CA 93101
(805) 963-3412

35mm SLIDE FROM YOUR PC

COMPUTER SLIDE EXPRESS converts graphic files produced on the IBM PC into brilliant 35mm color slides with color resolution 400% better than your monitor. Leave your printouts behind. Use high resolution color slides up to 4000 line. COMPUTER SLIDE EXPRESS $9/slide.
VISUAL HORIZONS
180 Metro Park
Rochester, NY 14623
(716) 424-5300

Languages

FINALLY! MODULES

Add class to your compiled BASIC programs with FINALLY! MODULES. Use pull-down WINDOWS, horizontal menus, pop-up help screens, input screen and directory managers. For use with FINALLY! Library and Quick Basic 2.0 or IBM compiler 2.0 30 day MoneyBack guar. Visa/MC/CK/MO. FINALLY! MODULES is $99.00 + $4.00 s/h.
Komputerwerk Inc. Dept PCT
851 Parkview Blvd.
Pittsburgh, PA 15215
(412) 782-0384

FORTRAN UTILITIES

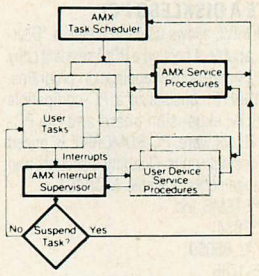
CROSS-REFERENCE UTILITY: Mainframe grade symbol x-ref listing for variables, subprogram calls and labels. Variable map shows type, length, alloc, scope, usage tag, etc. All FORTRAN 77 compilers. $49.95 + $2.50 s&h. UTILITY LIBRARY: Assembly language routines for screen cursor, keyboard, time, sound, etc. MS/IBM and IBM Pro/RM FORTRAN compilers. $39.95 + $2.50 s&h. IBM PC w/DOS 2.0+. Visa/MC/MO/check (2 wks).
PJN International
P.O. Box 201363
Austin, TX 78720
(512) 837-2888

Operating Systems



Real-Time Multitasking Executive

- No royalties
- Source code included
- Fault free operation
- Ideal for process control
- Timing control provided
- Low interrupt overhead
- Inter-task messages



Options:

- Resource Manager
- Buffer Manager
- Integer Math Library
- Language Interfaces : C Pascal PL/M Fortran
- DOS File Access : CP/M-80 IBM PC DOS

AMX is TM of KADAK Products Ltd
CP/M-80 is TM of Digital Research Corp
IBM, PC DOS are TM of IBM Corp

AMX for	8080	$ 800 US
	8086	950
	6809	950
	68000	1600
	Manual (specify processor)	75

KADAK Products Ltd. (604) 734-2796
Telex: 04-55670
206–1847 W. Broadway, Vancouver, B.C., Canada V6J 1Y5

CIRCLE 380 ON READER SERVICE CARD

Programmers Tools

ATTENTION TURBO PASCAL USERS!
Crash the 64K Barrier

Try **TURBO PACKAGE** now!
90 day money back guarantee!

Modular Programming!
Promotes **REUSE** of working **CODE**
CUTS development **TIME**
IMPROVES system **RELIABILITY**
SIMPLIFIES program **MAINTENANCE**

FILL 640KB with code/data any way you want
VERY FEW CODE CHANGES.
FASTER than chaining or overlaying

SUPERMATH, FREE!
With purchase of Turbo Package
40 plus LONG **(32-bit math)** routines
Faster than real - big enough for $.
ASM coding insures top performance

Just $49.95 (in TX add tax)
Visa/MC (no shipping chg)

Write or call for more information

CONVERSATIONAL COMPUTER SYSTEMS
5371 Verbena Rd.
San Antonio, TX 78240
Phone: (512) 692-0353

CIRCLE 395 ON READER SERVICE CARD

P-tral
BASIC TO PASCAL TRANSLATOR

Translate your BASIC source programs to Pascal source. P-tral, now available for the IBM PC and compatibles, will translate MS–BASIC/BASICA to Turbo Pascal (Req Dos 2.0 or later w/ANSI.SYS).

P-tral supports the translation of business, scientific, graphics and game software. And when necessary, P-tral interacts with you to pick out subroutines, name subroutines and rename variables which don't fit Pascal criteria. It will also prompt for fixed dimensions for dynamic arrays.

LIST PRICE: $179.

(212) 206-6490/924-0576
WOODCHUCK INDUSTRIES
340 WEST 17TH STREET (#2B)
NEW YORK, NY 10011

CIRCLE 398 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

ROMable CODE on PC!

PCLOCATE allows PC users to develop ROM-based software from MS-DOS "Exe" files. The user specifies the physical location of all segments. Output files are compatible with most PROM programmers. PCLOCATE supports the 8086, 8088, 80186, 80188, and 80286 processors. MC/VISA.
ALDIA SYSTEMS INCORPORATED
P.O. Box 37634
Phoenix, AZ 85069
(602) 866-1786

END YOUR FRUSTRATIONS—
MASTER YOUR SCREENS WITH FORMIX™

Developing, testing & changing screens is tedious, frustrating, time-consuming work.

Until now.

With FORMIX, you can trouble-shoot before writing one line of code. You can even prototype the entire system with actual screens—complete with data entry. Little changes in screen design are little effort. So are major changes.

Our advance panel-oriented system allows you to quickly develop complex screens that integrate several panels or windows.

Plus, FORMIX has an on-line help system. You just write the application help text. FORMIX handles the rest.

In short, FORMIX simplifies screen design and slashes the cost of program development. Let FORMIX handle the tedious, error-prone programming aspects while you concentrate on solving today's application problems—and eliminating a lot of tomorrow's.

And there's more. Contact us for details. We're ready to prove the power, flexibility & simplicity of FORMIX.

MSI
An Expression of Quality
Master Computer Systems, Inc.
FORMIX Division
9531 West 78 Street
Eden Prairie, Minnesota 55344
612/944-5220
FORMIX interfaces with ADA, Assembler, Basic, C, Cobol, Fortran & Pascal

GENSCREEN FOR MS-COBOL

Cobol Source Code Generator for generating the screen section and data division cobol source code for Microsoft and IBM PC cobol. Screen Image Text files are run through GENSCREEN to produce all of the source code for your screen in less than a minute. Super fast programmer productivity tool $69.99.
Personal Computer Development Corporation
P.O. Box 8556
Warwick, R.I. 02888-8556
(401) 333-8704

VERSION CONTROL SYSTEM

TLIB™ stores ALL versions of your source in ONE compact library file, even with hundreds of revisions. Updates (deltas), 5-7 times faster than Unix SCCS. Date & comments for each version, easy retrieval. LAN-shared libraries. Free public domain MAKE (with source) by Landon Dyer. DOS 2.x/3.x $99.95 $3 s/h VISA/MC.
Burton Systems Software
P.O. Box 4156-TJ
Cary, NC 27511-4156
(919) 469-3068

PASCAL-to-C TRANSLATOR

Industrial strength conversion from Turbo, Microsoft, UCSD, MT+, Apollo, Macintosh, and other Pascals to K&R C. Handles nested procedures, intrinsic functions, separately compiled units and modules, all data types including long integers.
Requires 512K IBM PC/XT/AT. Send up to 500 lines of Pascal and we will convert it for FREE. Site licensing from $5,000. Conversions 50 cents/line.
TGL Inc.
27096 Forest Springs Ln.
Corvallis, OR 97330
(503) 745-7476

TURBO FORMS

Bullet-Proof user data entry. Unlimited character & field level data verification. Create & edit forms for data entry & display without recompiling source code. Flexible formatting with graphics, windows, colors & display attributes. IBM PC & compatibles. One of PC Magazines "14 HOT TURBO UTILITIES." $39.95 including S&H. MC/VISA or C.O.D.
GREAT LAKES SOFTWARE SYSTEMS, INC.
2510 Capital Ave. SW Suite 203
Battle Creek, MI 49015
(616) 962-2017

SCREEN MANAGER

SAVE TIME! Powerful Screen Designer and Memory Resident Screen Manager Increases Programmer Productivity! Interfaces to most languages. BASIC, FORTRAN, COBOL, C, PASCAL, PLM86, ASM. Not a Code Generator! No Royalties.

$125 VISA/MC

The West Chester Group
P.O. Box 1304
West Chester, Pa 19380
(215) 644-4206

FREE DEMO DISK
CIRCLE 383 ON READER SERVICE CARD

Fortran Addenda '86

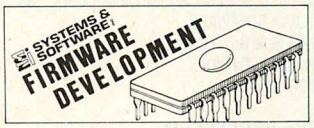
Libraries for graphics and friendly/interactive programs. ASMUTIL2: Total PC control; printers (3), CRTs (2), disks, FULL keyboard, strings, high-speed gets/puts, line/box, fills tile painting, CGA/EGA/Hercules graphics. BUTILE 2: Input wordprocessing/editing, non-overflowing formats, window management ... 100 easy to program, "smart" routines + defaults/toggles. 170 pg. manual & annotated samples. $95 alone; both $165. Specify compiler and version.

IMPULSE ENGINEERING

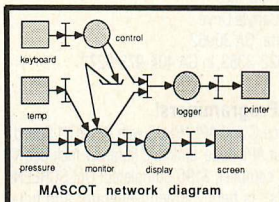
IMPULSE Engineering, B.R. Strong, Jr.
P.O. Box 3540
San Francisco, CA 94119-3540
(415)788-4611

FIRMWARE PRODUCTION ON PC

LINK&LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by C, PL/M compilers & MASM. Provides full control of segments placement anywhere in memory. Supports output of INTEL hex file for PROM programmer, absolute object file for symbolic debugger & ICE, and MS-DOS EXE file. Includes an INTEL compatible linker, locator, librarian and hex formatters. $350.



Systems & Software, Inc.
3303 Harbor Blvd., C11
Costa Mesa, CA 92626
(714) 241-8650



MASCOT network diagram

Andyne COMPUTING LIMITED

Suite 202, 544 Princess St., Kingston, Ont. Canada K7L 1C7 (613) 548-4355

PCMASCOT

Real-Time Multitasking Kernel for the IBM PC

- Supports MASCOT modular real-time design methodology
- Extensive built-in debugging facilities
- Shared memory for intertask communication
- Synchronization and mutual exclusion
- Modular design and implementation approach allows unit and sub-network testing
- Can use DOS DEBUG with application
- Can access all PC-DOS facilities
- C language interface (specify compiler)
- Device drivers may be written in C
- No royalties
- $795 includes software, manual, support, updates

CIRCLE 391 ON READER SERVICE CARD

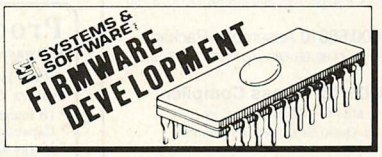
PC CROSS-ASSEMBLERS

Up to 10,000 lines per minute! Fast X-ref and Linker plus Macros and Librarian. Generates HEX, TEKHEK, S-records, and .OBJ output records. Over 40 micros and XENIX, MS DOS, CPM 80 and ISIS versions. Accepts MOTOROLA and INTEL directives and Mnemonics.
RELMS™
P.O. Box 6719
San Jose, CA 95150
(408) 265-5411

BOOSTERS V2.0 IS HERE!

Tools for Turbo Pascal programmers who need the speed and efficiency of inline code. 70+ string, video, and DOS routines—incl. Exec. V2.0 also incl. powerful new SCREEN GENERATOR, DOS SHELL, and many example programs. All Pascal and assembler source, manual, update notices. No Royalties. $40 + 4% GA tx. Visa/MC.
GEORGE F. SMITH & COMPANY
609 Candlewick Lane
Lilburn, GA 30247, (404) 923-6879

LINK & LOCATE



LINK & LOCATE enables PC users to produce ROM-based firmware for 8086/87/186 from object files generated by popular C compilers, such as from Wizard, Microsoft and Lattice, and MASM assembler from Microsoft. Provides full control of segment placement anywhere in memory. Supports output of Intel HEX file for PROM programmers, Intel OMF absolute object file for symbolic debuggers and in-circuit emulators. Includes Intel compatible linker, locator, librarian and hex formatters. $350.

Systems & Software, Inc.
3303 Harbor Blvd., C11, Costa Mesa, CA 92626
Phone (714) 241-8650 FAX (714) 241-0377 TWX 910-695-0125

CIRCLE 384 ON READER SERVICE CARD

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

PROGRAMMERS TOOLS

MS-COBOL SCREEN/DATA DIV.

MSCREEN generates Screen Section code for MICROSOFT/IBM COBOL. Paint/Edit screens. No other editor needed. Select from complete set of attributes for each field. No field terminators. Many other features! COBWORK generates Data Division code for MICROSOFT/IBM/COBOL. $85.
TAJEVA SOFTWARE
6064 Belle Grove Cove S.
Memphis, TN 38115
(901) 365-4692

PRE-PROCESSOR

Add custom features to any language: longer identifiers, opcode, register and operator synonyms, nested macros, etc. C Source Code included. Not copy protected. OK to share. $19.95 + s/h. MC/VISA.
SUPERTECH
11410 NE 124 St., #6143
Kirkland, WA 98034-4399
(206) 488-9253

Quelo® 68000

Software Development Tools

Quelo Assembler Packages are **Motorola compatible.** Each package includes a macro assembler, linker/locator, object librarian, utilities for producing **ROMable code,** extensive indexed typeset manuals and produces **S-records,** Intel hex, **extended TEK hex, UNIX COFF** and symbol cross references. **Portable source** written in "C" is available. It has been ported to a variety of mainframes and minis including **VAX.**

68020 Assembler Package
For CP/M-86,-68K and MS/PC-DOS . $ 750

68000/68010 Assembler Package
For CP/M-80,-86,-68K and MS/PC-DOS . $ 595

68000 **"C" Cross Compiler**
For MS/PC-DOS by Lattice, Inc.
With Quelo 68000/68010 Assembler Package $1095
With Quelo 68020 Assembler Package . $1250

Call **Patrick Adams** today:

Quelo, Inc.
2464 33rd W. Suite #173
Seattle, WA USA 98199
Phone 206/285-2528
Telex 910-333-8171

COD, Visa, MasterCard

Trademarks: CP/M, Digital Research; MS, Microsoft Corporation; Quelo, Quelo, Inc.

CIRCLE 382 ON READER SERVICE CARD

BASIC + StruBAS

Developing serious applications in compiled BASIC? It's easier with StruBAS v2.0 tools complementing QuickBASIC and IBM BASIC 2.0 with extended structured code, screens, menus, native ISAM, Btrieve interface, and subroutine object library. $495 single, $1495 site license. VISA/MC. Not copy protected.
Laney Systems Inc.
3 Office Park Dr., Suite 100
Little Rock, AR 72211
501-225-7755

True Shell for BASIC

SHELL any other program or batch file, including other compiled BASIC programs and the BASIC interpreter. Requires DOS 2+ and IBM (Ver 1 or 2) or Microsoft compiler (QB1, QB2 or 5.36). QB2 requires DOS 3+. Only $29.95 + $3 s/h. MC/VISA/COD OK. 30 day money-back performance guarantee.
MicroHelp, Inc.
2220 Carlyle Drive
Marietta, GA 30062
800-922-3383. In GA 404-973-9272

APL Programmers!

Interface C and APL*PLUS with APL2C™! Speed up your APL code. Link to C libraries. Includes K & R C compiler. $195 Complete. FULLSCREEN Panels™ is here! Screen Generator and full-screen processor for the APL environment. Pop-ups, panels, menus, scrolling fields NO ROYALTIES, $150.
Lauer Software
PO Box 728
Newtown, PA 18940-0728
(609) 921-6249

BASIC Base 007

BASIC database library including menues, passwords, program generator, query, screen control, data record control, index commands for add, delete, find, find next, find last. $15 demo with disk manual and compiled database software. $99 development system with library source code and printed manual. $165 for compiled & all 4,000 + lines of BASIC code.
Application Micro Computers, Inc.
1663 Bachan Ct.
Reston, Va. 22090
(703) 471-1471 - 3:00 to 9:00 P.M.

Productivity Tools

Software Revision Management System™

SRMS is a full featured version control system featuring:
* 10 Integrated Utilities with user shell
* Capability for hundreds of versions/library
* Merge utility resolves parallel effort conflicts
* Report Generation Utilities
* Typeset documentation and much more!

Version 3.0 $185

QMAKE™
* Program generation utility patterned after UNIX make to aid in rebuilding sytems
* Recompiles only routines necessary
* Support for macros and multiple entry points
* Integrates fully with SRMS

Version 1.2 $99

TXTTOOLS
* QDIFF - Windowed File Difference Utility
* QSE - Quilt Stream Editor
* QSRCH - Like UNIX GREP

$85

Quilt Computing
7048 Stratford Road
Woodbury, MN 55125
(612) 739-4650

CIRCLE 396 ON READER SERVICE CARD

FRUSTRATED WITH ASSEMBLER

Unique shareware program adds structure to assembly language. IBM/MASM compatible. Develop and debug faster with structured concepts. IF-THEN-ELSE, SELECT-WHEN-OTHER, DO WHILE/UNTIL/INCREMENTAL and combinations. LEAVE/LEAVEALL loop exits, and file includes. Nest 16 deep. $6 disk, $25 registration.
MICHAEL T. HOLLAND
8808 Boulder Lane
Little Rock, AR 72207
(501) 224-2749

Turbo Pascal Programmers: 15 MINUTES = 200 HOURS! with new *turboMAGIC* code generator.

Input forms and help windows up to 66 lines long. Scrolling within framed windows for data-entry or help. Pop-up menus. Pull-down menu systems. And **much** more! Just $99. 30-day money-back guarantee. Requires 256K IBM PC compatible. Orders: 800-225-3165

turbo magic

Sophisticated Software
6586 Old Shell Road
Mobile, AL 36608
205-342-7026

CIRCLE 397 ON READER SERVICE CARD

"NEW" BIT-LOCK® SECURITY

Piracy SURVIVAL ">4" YEARS proves effectiveness of powerful multilayered security. Uses rapid decryption algorithms and small reliable port for transparent security device. NOW AVAILABLE for PARALLEL or SERIAL port. NEW KEY-LOK™ security device available at HALF-PRICE.
MICROCOMPUTER APPLICATIONS
7805 S. Windermere Circle
Littleton, CO 80120
(303) 798-7683 or 922-6410

PUBLIC DOMAIN

TURBO PASCAL™ SOFTWARE $6

Write or call for information about:
- Systems & applications development tools
- Programs for home and business
- Communication tools & applications
- Games in specialized applications
- Scientific/engineering programs & routines
- Graphics including animation tools

TURBO S.I.X.
P.O. Box 8373
Waco, TX 76714
(817) 753-2182

Tech Marketplace, the home of the power buyer.

NEW **PUBLIC DOMAIN** LISTING
13,000 **MS DOS PROGRAMS** with brief descriptions, 52 pages, **$4.** Also available on disks for $10 including search program. This months special set 5 disks $2 including p+h. 90 programs including Mandelbrot Set Images, CalTech utilities, advanced Lotus tutorial, artificial art, Freecalc V2, Genealogy V4. Send your card + $4 to or call: The Public Domain Software Co.
THE PUBLIC DOMAIN SOFTWARE COPYING COMPANY
33 Gold Street
NYC, NY 10038
800-221-7372 • NY 212-732-2565

TURBO PASCAL $2/disk

TSS is a BBS-by-mail, no modem needed (long distance is more $$$ than mails)! 60+ disks of Pascal files. Most incl. source code. All files compressed. Membership fee ($25) incl. free starter pkg. and 2 FREE disks with 1st order. Non-members $7/disk. Cat. list $5. VISA/MC/COD (s/h extra) (data) 617-545-9131
TURBO SOURCE SEARCH
P.O. BOX 876
SCITUATE, MA 02066
(voice) 617-545-6677

SCIENTIFIC

SCI/ENG GRAPHICS

OMNIPLOT [S] (screen graphics) & OMNIPLOT [P] (plotter driver) provide integrated engineering/scientific 2-D & 3-D graphics with NO PROGRAMMING! Menu-driven, flexible, professional. Choice of formats: tabular/line, contour, bar, pie, 3-D wire frame & much more! OMNIPLOT [S] $195. Add OMNIPLOT [P], both $295.
MICROCOMPATIBLES, INC.
301 Prelude Dr. Dept. J
Silver Spring, MD 20901
(301) 593-0683

8087 FFT/VECTOR PROCESSING

The VECTOR87 library is written in assembler, includes 60 routines to speed up your number-crunching programs. Uses 80(2) 87 extensively. PC 1K real FFT takes only 1.2 sec. Versions for Fortran (MS, RM, Lahey), C (MS, Lattice), Turbo Pascal -87. $150 per version with source, no royalties. Write for technical information.
VECTORPLEX Data Systems Ltd.
136-100 Maitland Place N.E.
Calgary, Alberta, Canada T2A 5V5
(403) 248-1250

DATA ACQUISITION & ANALYSIS

*MEASURE for data acquisition directly to Lotus 1-2-3*FOURIER PROSPECTIVE II advanced signal digital analysis *Lotus Manuscript & technical document preparation system*PRIME FACTOR FFT subroutine library. Call Turbo Pascal, C, Fortran, Basic. Up to 65,520 data-points. 2D available*Turbo Pascal from Borland *TELEVISION for Image Communications *8087 Coprocessors, all varieties*Dash-16A/D converter board from MetraByte.
ALLIGATOR TECHNOLOGIES, INC. P.O. Box 11386 Costa Mesa, CA 92627 (714) 662-0660

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

SCIENTIFIC

NUMERICAL C SOFTWARE

Computationally stable numerical routines for scientific C software developers. LINLIB contains all the basic vector and matrix routines solutions to equations, LU, QR, Cholesky factors of matrices, least squares solutions. LINLIB has splines, B-spline routines, spline interpolation, spline approximation of data. $150.
INFORMATION AND GRAPHIC SYSTEMS
15 Normandy Court
Atlanta, GA 30324
Call (404) 231-9582

SECURITY DEVICES

SMART COPY PROTECTION

Attention Software Developers, are you tired of Copy Protection that:
-is NOT transparent to the user.
-does not allow backups.
-requires I/O plugs or special media.
-doesn't support hard or cartridge disks.
-makes you pay for every disk protected.
-requires source code changes.
-can be beaten by hardware copy boards.
If so, EVERLOCK can solve these problems for only $495. Free info & demo disk available.

Az-Tech Software, Inc.
426 Grandview
Richmond, MO 64085
(816) 776-8153

SECURE AT/XT/PC

Control system access, data access! FiXT/S. Control system boot for most popular XT/PC hard disk controllers. Feature for AT-and-XT-compatible HD controllers segments hard disk by volumes, controls access with passwords, supports hard disk expansion. $80 $120 + $3 shpg. plus CA tax.



Golden Bow Systems
2870 Fifth Ave. Suite 201
San Diego, CA 92103
(619) 298-9349

Multi-User Security

File access by user at the directory or sub-directory level. Protection from unauthorized formatting of specified drives by user. User transparent. Accounting reports by user and category. PC-LOCK $90.
ONTAR Corporation
129 University Road
Brookline, MA 02146-4532
617-739-6607

STATISTICS

RATS! VERSION 2.0

RATS, the best selling Econometric software package now includes daily & weekly data, a new, easier to use 500-page manual, & many advanced features. Use RATS for time-series & cross-section regression, including OLS, ARIMA, VAR, logit, & probit. IBM PC or compatible. $200. VC/Visa. Call for brochure.
VAR Econometrics, Inc.
P.O. Box 1818
Evanston, IL 60204-1818
1(800) 822-8038

P-STAT®

Full mainframe package for IBM PC/XT/AT & compatibles. Combines data & file management, data display, statistical analysis, report-writing & survey analysis in a single package. 4GL programming language, online HELP, menu or command driven with interactive EDITOR. $95 demo and Site License available.
P-STAT Inc.
471 Wall Street, P.O. Box AH
Princeton, N.J. 08542
Telephone: 609-924-9100
Telex: 466452

STATISTICAL FORECASTING

AUTOBOX, AFSEZF, AUTOBJ, BOXX, MTS and SIMBOXJ-a complete line of programs for Box-Jenkins time series analysis and forecasting. Combine the ultimate in sophisticated forecasting procedures with unparalleled ease of use. Call or write for more information-find out why our users are our best reference!
AUTOMATIC FORECASTING SYSTEMS, INC.
P.O. Box 563 Dept. T
Hatboro, PA 19040
(215) 675-0652

If you can buy only one statistics and forecasting package, choose the best.

StatPac Gold™

Call for free descriptive brochure:
1-800-328-4907

WALONICK ASSOCIATES, INC.
6500 Nicollet Avenue S. Minneapolis, MN 55423
(612) 866-9022
CIRCLE 393 ON READER SERVICE CARD

To place your ad in **Tech Marketplace** call **212-503-5115.**

All the Stats You Need

BMDP offers the most comprehensive collection of programs for PCs & mainframes. BMDP has all the statistics you need. From the simple to the most sophisticated. Advanced techniques include time series, survival analysis, log-linear modeling & more! Hard disk req'd. Call for complete program details & prices.

BMDP STATISTICAL SOFTWARE

BMDP Statistical Software, Inc.
1440 Sepulveda Blvd.
Los Angeles, CA 90025
(213) 479-7799

TAXES

Where Does the Time Go?

TUSKER knows! TIME & USAGE KEEPER logs and reports your computer time; meets and exceeds IRS requirements for proving tax deduction.
- Define your own business uses
- 6 reports in any date range for any printer
- Log non-computer time too!

DOS 2.0+. $88. Free brochure. $4 demo disk.
Craig Banning
Route 3, Box 317
Big Pine Key, FL 33043
(305) 872-3817

TERMINAL EMULATION

BARR/HASP INTELLIGENT RJE WORKSTATION

Hardware and software communications package for IBM PC, XT and AT. Simultaneously transmits data to host and receives output directly to MVS/JES2, MVS/JES3, VS/RSCS, and CDC/NOS, bypassing TSO and CMS. Emulates IBM 3777-2 and HASP on IBM 360/20. Line speed: 1,200 to 19,200 baud (56,000 bps on AT). Supports multiple high-speed printers beyond 2,400 lpm. (6,000 lpm on AT). Features: concurrent DOS, LAN support, printer forms control, plotter support, unattended operation, easy installation. $1,290 includes Hardware & Software.

BARR

BARR SYSTEMS, INC.
2830 NW 41st Street, Building M
Gainesville, FL 32606
(800)-BARR-SYS/(904) 371-3050

UTILITIES

TallScreen—DOS POWER

Natural extension of DOS. Scroll back through screen output, edit text on full screen, mark blocks to printer or file, recall commands & directories, enter multiple commands, capture screens from application programs, create user profiles. Solid tech support. PC MAG & PC WORLD calls TallScreens a Real bargain at $49.95. VISA/MC

QUALITAS™

Qualitas, Inc.
8314 Thoreau Drive
Bethesda, MD 20817
(301) 469-8848

SAVE THAT SCREEN!

Do you immediately reach for the PrtSc key to save screen info? What a waste of time and paper! Now, SCREENSNAP™ lets you save and recall up to 9 screens at the touch of a key. Friendly with other resident programs but unlike some it is compact; will run in as little as 5K. Also includes useful utilities to save and recall from files, programmer's interface and sample code. Build your own help screens with your text editor, then save and recall them with SCREENSNAP. $39.
Programming ARTS
P.O. Box 219
Milltown, NJ 08850
Call 800-443-4160; NJ (201) 846-7242

DOCUMENTATION MANAGER

Create and maintain manuals - procedure manuals, program documentation / system user manuals, etc. * Edit files with the excellent Norton Editor (included) * Save User Defined configuration * Save screen dumps to files * Variety of Print Options *
$69.95 complete MasterCard/Visa

PHENIX

PHENIX HOSPITAL SYSTEMS
1616 Palm Avenue
Deland, FL 32724
(904) 736-1132

AT/XT/PC HARD DISK EXPANSION

"Replace hard disk with a bigger one, or add a second drive! Vfeature BREAKS THE 33 MBYTE BARRIER on standard AT, XT, and compatible hard disk controllers. Includes multiple volumes, security features, selectable clusters, keyboard lock. $80-$120 + $3 shipping + CA Tax"



Golden Bow Systems
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

UTILITIES

LIMSIM

Expanded Memory Simulator for the PC/AT and compatible 286 machines. Use the extended memory you already have as Lotus style Expanded Memory. Fully supports EMS version 3.2. Requires 70k of conventional memory. $50 ($75 with assembler source) plus $5 s/h. 30 day money back guarantee.
Larson Computing
1556 Halford Ave. #142
Santa Clara, CA 95051
(408) 737-0627

DISK UPGRADE BIOS for ATs

DUB-14 overides AT Drives Table to allow any compatible drive to be attached and fully used on the standard AT controller. Two ROMs plug into empty sockets on system board. Includes complete Set-Up routine and low-level format facility. Works with UNIX, XENIX, other OS and networks. $95 + $3 shpg. CA tax.



GOLDEN BOW SYSTEMS
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

HARD DISK EXPANSION!

Disk Manager allows the installation of any ST506 hard disk on PC,XT,AT and compatibles. Volumes up to 256mb! Menu driven/auto install, compatible w/ all vers of MS/PC DOS (does not modify DOS), up to 16 volumes, easy to use! $125+ ship. Ask about Novell product! Dealer inquiries invited.

ONTRACK COMPUTER SYSTEMS INC.

Ontrack Computer Systems, Inc.
6222 Bury Drive
Eden Prairie, MN 55344
(612) 937-1107

DISK ACCELERATOR V2.0

DiskCache speeds up your hard disk access. Disk caching and ram disk in one package. Ram disk shares cache space. Transparent, flexible, configurable, no h/w changes. RAM, EMS, and AT extended memory versions incl. Not copy protected. VISA, MC, volume discounts. No PO's w/o prior approval. $49.00
Datamorphics Ltd.,
P.O. Box 820
Stittsville, Ontario, Canada KOA 3G0
Or call (613) 836-2670

FILE PRINT MANAGER

GLISTER™
★ Use DOS wildcards to build a list of up to 100 files to print
★ Save/restore file lists
★ Restart a file on any page after a printer jam
★ Print multiple copies
★ Control: margins, line/page length, spacing, user-formatted header/footer lines and more
★ Prints files as fast as printer is capable $49
Programming ARTS
P.O. Box 219
Milltown, NJ 08850
Call 800-443-4160; NJ (201) 846-7242

VCACHE GETS YOUR DISK MOVING!

Hard disk accelerator increases speed of cartridge and fixed disk operations using memory caching to eliminate repetitive disk access. Allocate up to 15Mb of extended or expanded memory, or .5Mb of standard memory for caching disk data. Includes diskette and screen accelerator modules. Automatic and transparent after installation. $65+ $3 shpg, CA tax.



GOLDEN BOW SYSTEMS
2870 Fifth Avenue, Suite 201
San Diego, CA 92103
(619) 298-9349

CHARACTER CUSTOMIZATION

CHARGEN! 3.0 works with the IBM/EGA to let you modify the character set, allowing many wordprocessors to display technical material, equations or other special characters. Requires DOS 2 x or 3.x, IBM Standard or Enhanced Graphics Adapter. $35+ $2 s/h (MN add 6%).
DK Micro Consultants
P.O. Box 6714
Minneapolis, MN 55406
(612) 722-0931

THE NEWMAN UTILITIES

50 utils includes help system below and disk + system utilities $19.95
EZRUN menu. Run 1-36 programs $19.95
CACHER. speedup disk access 10X $19.95
HELP system for DOS 3.1 + add your own $9.95
All $45, $2 demo, 15 day MB guar., $2 Ship
NEWMAN COMPUTER
2 Briar Mills Drive Suite 2-A
Bricktown, NJ 08724
(201) 458-5169

Quaid Analyzer

the tool that created CopyWrite

Now you can debug your own programs with a professional quality debugger - the one that unraveled every form of copy-protection used on the PC.

With the Quaid Analyzer, you can:

- ☐ See occurences of any interupt, with its meaning shown on the screen.
- ☐ View memory as text or instructions, scrolling as easily as you do with an editor.
- ☐ Run until a memory location or I/O port is changed.
- ☐ Protect your hard disk from accidental destruction.
- ☐ Analyze software without the source, even when it uses countermeasures to thwart tracing.
- ☐ See all stages of the boot load.

We kept the Quaid Analyzer off the market to avoid helping publishers with copy-protection. Now that copy-protection is gone, we can sell it to you.

The Quaid Analyzer is a software tool occupying 100K bytes. It runs on any IBM PC and most MS-DOS systems without hardware modification.

Quaid Software Limited

$99 U.S. To order Call **(416) 961-8243**

All orders shipped at our expense within a day. All major credit cards accepted.

or write to:
45 Charles St. East
Third Floor, Dept. 602
Toronto, Ontario. M4Y 1S2

Ask about Disk Explorer the program that takes over where Quaid Analyzer leaves off.

CIRCLE 387 ON READER SERVICE CARD

PC Tech Journal Marketplace:

Kathryn J. Cumberlander
Classified Advertising Director

Daniel L. Rosensweig
Classified Sales Manager

Lisa B. Stick
Account Manager

Call: (212) 503-5115

TECH MARKETPLACE

THE COMPREHENSIVE GUIDE TO PRODUCTS AND SERVICES FOR THE MS DOS MARKET

SOFTWARE/UTILITIES

XT/AT HARD DISK DIAGNOSTICS!

Disk Manager Diagnostics performs extensive tests on your ST412/506 hard disks. Areas tested are: Controller, data write/read, seek test, automatic error correction(ECC), random reads and media defects. Interactive help. Excellent error detection and isolation. $49.95 + ship. VISA/MC accepted.

ONTRACK COMPUTER SYSTEMS INC.

Ontrack Computer Systems, Inc.
6222 Bury Drive
Eden Prairie, MN 55344
(612) 937-1107

MAKE YOUR PC SEEM LIKE AN AT!

MAKE YOUR AT SEEM LIKE A DREAM MACHINE!

ANSI-CONSOLE™

The Integrated Console Utility™

FAST, POWERFUL ANSI.SYS REPLACEMENT

For the IBM-PC, AT, and clones
New Version 2.00 is MUCH FASTER

Now blink free scrolling on CGA!
Now use EMS for scroll recall!
New option menu program!

- Speed up your screen writing
- Extend your ANSI.SYS to full VT100
- Scroll lines back onto screen
- Save scrolled lines into a file
- Add zip to your cursor keys
- Free your eyes from scroll blinking
- Easy installation
- Get 43 line EGA support
- Over 50 useful options

"The psychological difference is astonishing" -*Lotus* June 85 pg 8.

"So many handy functions rolled into one unobtrusive package" -*PC-World* Feb 86 pg 282.

"The support provided by the publishers is extraordinary." -*Capital PC Monitor* May 86 pg 25.

"...the best choice for improving your console..." -**Capital PC Monitor** June 86 pg 282.

460p Manual (w/slip case) and software diskettes $75.

Satisfaction Guaranteed! Order Yours Today!

HERSEY MICRO CONSULTING
Box 8276, Ann Arbor, MI 48107
(313) 994-3259 Visa/MC/Amex
DEALER INQUIRIES INVITED

CIRCLE 388 ON READER SERVICE CARD



AT's DON'T NEED 360KB DRIVES

The 1.2MB drive has long been known to **READ** but **NOT** reliably **WRITE** on 360KB floppies. With **"CPYAT2PC"** 1.2MB drives **CAN** reliably **WRITE** 360KB floppies saving a slot for a second hard disk or backup tape. **"CPYAT2PC"** (Not Copy Protected) offers the preferable **SOFTWARE SOLUTION.**

- **NO** software or hardware modification
- A 360K drive is **NOT** required
- **"CPYAT2PC"** program **MAY** reside on hard disk
- Runs on IBM PC/AT and COMPATIBLES i.e. Compaq Deskpro 286/386, AT&T 6300+, HP Vectra, Sperry PC/IT, Tandy 3000

Only $79.00 + $4.00 S/H VISA, MC, COD, UPS-B/R
ORDER TOLL FREE 1-800-621-0851 XT777
TELEX EZLINK 62873089
Dealer Inquiries Invited
MICROBRIDGE COMPUTERS
655 Skyway #125
San Carlos, CA
CA 415-593-8777
NY 212-334-1858

CIRCLE 386 ON READER SERVICE CARD

Compress your data into 1/10th the space!

Introducing ARC. It's used to create and maintain data file archives for computers operating under any DOS system.

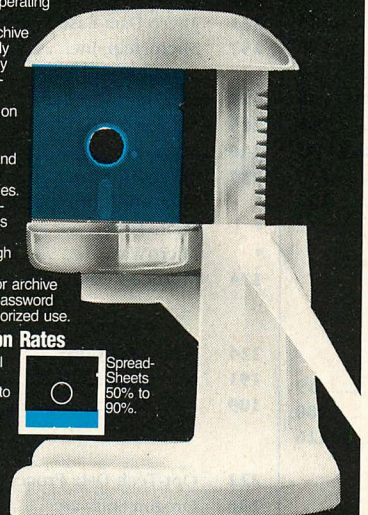
But it does something that other archive and library utilities can't. It automatically squeezes the files being saved so they take up less space. Like a can of concentrated orange juice.

From 20% to 90% less, depending on the kind of data being saved!

So there's more room to store data, no matter what media it's stored on! And that's like giving a shot of vitamin C to your savings on equipment and supplies.

This compressed data can be transmitted over telephone lines in a lot less time than it takes to transmit uncompressed data. So you can beat the high cost of phone bills to a pulp, as well.

ARC has a full range of functions for archive creation and maintenance. Including password encryption to protect data from unauthorized use.



Typical Compression Rates

Program files 20% to 30%.
ASCII text 50% to 60%.
Spread-Sheets 50% to 90%.

sea

System Enhancement Associates • 21 New Street, Wayne, NJ 07470 • (201) 473-5153

CIRCLE 389 ON READER SERVICE CARD

Tech Marketplace . . . the comprehensive guide to products and services for the MS DOS market.

MISCELLANEOUS

BAR CODING

$99 BAR CODE READERS

We need Distributors & OEMs worldwide. Our readers are IBM PC/XT/AT & Tandy 1000/2000 keyboard compatible, convertible to RS232 Interface, have auto code distinction, need no additional software and are availabe from assembled board to fully packaged units. From US $99 plus wand in modest OEM quantities.
ASP MICROCOMPUTERS
P.O. BOX 259, CAULFIELD EAST 3145
VICTORIA, AUSTRALIA
PHONE 011 61 3 5000628 (note time difference)

BAR CODE READERS

- IBM, AT&T, Link, Kimtron, TeleVideo,
- Alloy, DEC keyboard interfaces or RS-232C
- Need others? Call.
- NO programming. Reads dot matrix
- Auto-recognition and single code decoding
- Reads Code 39, UPC A/E, Codabar & 1 2 of 5
- Units in stock, 2 year warranty
- Bar code printing software, call for info

PERCON®

PERCON, Inc.
2190 W. 11th
Eugene, OR 97402
(503) 344-1189

The BEST BAR CODE READER for the IBM PC & AT $595

Simple & quick installation
No additional software or port
Metal wand & case

Also available:
Bar code printing software
Magnetic stripe readers
Units for other computers & terminals

TPS ELECTRONICS
4047 Transport Street
Palo Alto, CA 94303
Telephone: 415-856-6833
Telex: (Graphnet) 371-9097 TPS PLA

CIRCLE 394 ON READER SERVICE CARD

PUBLICATIONS

AN INVITATION TO STEAL

Bad copyright—or none at all? Your software may be public domain. Protect your work before you show it. Learn how to copyright software inexpensively and effectively. Software Copyright Guide. $6 ppd.
Innovation Press
Dept. 112
Box 351
Highland, IL 62249

READER SERVICE NUMBER	ADVERTISER	PAGE
103	ACS International	149
116	Advanced Logic Research	Cover 3
153	Aker	158
106	Aldebaran	179
118	American Small Business	184
136	Arity Corporation	136
206	ATI	155
122	Arrix	111
*	AST Research, Inc.	62
126	AST Research, Inc.	165-167
203	Atron	8
249	Atron	16
238	Atronics	89
165	Attachmate	151
105	Barrington Systems	117
147	Barr Systems	64
*	BC Associates	194
104	Blaise Computing	19
*	Borland Int'l.	Gatefold Cover
254	Borland Int'l.	1
117	Burr-Brown Corp.	173
*	Business Engrg. Systems	192
111	Catamount Corp.	125
188	Century Software	148
114	ComCal	56
*	Creative Programming Consultants	86
145	Cresent Software	168
167	Crosstalk Communications	Back Cover
261	Custom Software Systems	138
217	CXI	137
107	Data Light	39
214	Desktop A.i.	180
123	Digi Data	126
131	Ecosoft	162
190	ESP Software	171
112	Evsan Co.	194
119	FairCom	97
*	Flagstaff Engineering	116
199	Galacticomm, Inc.	22
*	Gimpel Software	124
113	Haven Tree Software Limited	20
132	Hawaiian Village	190
149	Ibex Computer Corp.	168
210	I-Bus Systems	174
134	Information Technologies	195
138	Information Technology	82
135	Innovation Computer	178
108	Innovative Data Tech	154
216	Intel Corp.	44 & 45
209	Interactive Microware	168
139	Isogan	112
258	KADAK Products	180
148	Kurtzberg Computer Systems	175
128	Lahey Computer Systems, Inc.	80
160	Lattice, Inc.	127
173	Lifeboat Assoc.	99
229	LOGITECH Inc.	113
125	Lugaru	172
110	Macmillan Software	153
263	Mansfield Software	162
207	Mark Williams Co.	119
146	MDS, Inc.	163
259	MetaWare, Inc.	180
211	Micro Data Base Systems	54
257	MicroHelp, Inc.	106
*	Microsoft Corp.	21
121	Microsoft Corp.	30 & 31
205	Microsoft Corp.	26 & 27
140	Microsoft Corp.	28 & 29
237	Microsoft Corp.	23
*	Microsoft Corp.	74
*	MicroTech	88
*	MicroWay	189
174	Mortice Kern	161
224	Nantucket Corp.	139
191	Norcom	164
109	Novell	176
222	Opt-Tech Data Processing	4
200	Oregon Software	58
185	Overland Data, Inc.	132
230	Paradise Systems	6 & 7
101	Paul Mace Software	130
*	PC Brand	140-143
159	PC Designs	191
150	PC's Limited	12 & 13
202	Peripheral Marketing	173
215	Perisope Company	5
175	Programmer's Connection	41-43
143	Programmer's Paradise	193
220	Programmer's Shop	46
162	Programmer's Shop	24 & 25
183	Proteus	192
166	Raima Corp.	11
157	Rainbow Technology	180
171	Ram Explosion	194
181	Rational Systems	109
137	Ryan-McFarland	187
*	Scantel Systems Ltd.	194
187	Scientific Endeavors	125
201	SoftCraft Inc.	2
186	Soft*Rite	108
142	Software Garden	93
196	Software Link	52
168	Software Masters	170
120	Software Merchants	123
189	Software Security	98
144	Software Solutions	100 & 101
130	Solution Systems	14
177	Stargate Technologies	168
198	Sterling Castle	150
176	Storage Dimensions	164
195	Summit Software	68 & 69
158	Sunny Hill Software	152
184	Sunny Tech Inc.	190
231	Systems & Software	134
194	Tall Tree Systems	35
197	Tall Tree Systems	37
155	Telebyte	18
156	Unify Corp.	133
169	Upper Bound Micro Computer	129
115	Vermont Creative Software	17
204	Video 7	135
124	Wells American	107

RS#	PRODUCT	ADVERTISER	PAGE
	IBM AND COMPATIBLE PC'S		
116	The PC2/286	Advanced Logic Research	Cover 3
126	AST PC	AST Research	165-167
159	GV 386	PC Designs, Inc.	191
150	Mail Order	PC's Limited	12&13
183	Compatible	Proteus Technology Corp.	192
124	A * Star	Wells American	107
	MULTIFUNCTION/MEMORY CARDS		
210	I-Bus IQ188 Parallel Processor	Ibus Systems	174
216	Inboard 386/AT	Intel Corporation	44&45
202	Fastcard	Peripheral Marketing, Inc.	173
197	RT Board	Tall Tree Systems	37
	GRAPHICS CARDS		
103	ACS 1000/ET286	ACS International	149
206	EGA Wonder	ATI Technologies	155
238	Professional Image Board	Atronics International Inc.	89
108	LEO	Innovative Data Technology	154
230	Auto Switch EGA	Paradise Systems	6 & 7
204	Vega Deluxe	Video Seven	135
	MASS STORAGE HARDWARE		
111	9 Track Tape System	Catamount Corporation	125
123	9 Track Tape System	Digi Data	126
*	9 Track Tape System	Flagstaff Engineering	116
149	9 Track Tape System	Ibex Computer Systems	168
185	Tapelinx/Finance	Overland Data, Inc.	132
231	Perstor 200	Systems and Software	134
155	Multifunction Storage	Telebyte	18
169	The Eagles Series	Upper Bound Micro	129
	PRINTERS-PLOTTERS		
194	J Laser Plus	Tall Tree Systems	35
	DATA ACQUISITION		
117	DMA Plug in Carrier Board	Burr-Brown Corporation	173
209	Catalogue	Interactive Microware	168
	MICRO-MINI MAINFRAME COMMUNICATIONS		
217	PCOX 5250 Prods.	CXI	137
189	The Block	Software Security	98
	LOCAL AREA NETWORKS		
165	3-N-1	Attachmate Corporation	151
149	9 Track Tape Systems	Ibex	187
138	Link Up, LAN Gateway	Information Technologies	82
186	LANscreen, LANbasic, LANbasic	Soft*Rite	108
	OTHER COMMUNICATION HARDWARE		
199	Galacticomm Breakthrough	Galacticomm, Inc.	22
109	LAN Report	Novell, Inc.	176
177	Strgate OC3000	Stargate Technologies	168
	OTHER COMMUNICATION SOFTWARE		
188	Term	Century Software	148
167	Crosstalk	Crosstalk Comm.	Back Cover
199	Galacticomm Breakthrough	Galacticomm	22
186	LANscreen, LANbasic, LANbase	Soft*Rite	108
196	Multi Link Advanced	Software Link	52
	LANGUAGES		
136	Prolog Compiler	Arity, Inc.	136
105	Clarion	Barrington Systems, Inc.	117
254	Turbo Pascal, Gameworks	Borland Int'l.	Gatefold Cover & 1
254	Turbo Database Toolbox, Prolog	Borland Int'l.	Gatefold Cover & 1
254	Turbo Editor, Editor Toolbox	Borland Int'l.	Gatefold Cover & 1
254	Turbo Graphix Toolbox	Borland Int'l.	Gatefold Cover & 1
131	ECO-C88-Microstat	Ecosoft Inc.	162
135	Marshal Pascal	Innovation Computers	178
128	F77L Lahey Fortran	Lahey Computer Systems	80
173	Best New Programs	Lifeboat Associates	99
229	Modula 2	Logitech Inc.	113
263	Personal REXX	Mansfield Software	162
207	Let's C	Mark Williams Co.	119
121	Quick Basic II	Microsoft Corporation	30 & 31
*	Language Newsletter	Microsoft Corporation	21
140	Microsoft C	Microsoft Corporation	28 & 29
237	Microsoft Assembler (MASM)	Microsoft Corporation	23
205	Microsoft Fortran	Microsoft Corporation	26 & 27
200	Pascal Compiler	Oregon Software	58
130	Brief	Solution Systems	14
195	Better Basic	Summit Software	68 & 69
	PROGRAMMER'S TOOLS		
106	Source Print	Aldebaran	179
122	Micro DCF	Arrix Logic Systems	111
203	PC Probe	Atron	8
249	Mini Probe	Atron	16
104	C Tools Plus	Blaise Computing	19
*	Vitamin C	Creative Programming Cons.	86
145	Quick Pak	Crescent Software	168
261	PC/VI	Custom Software Systems	138
107	C Compiler	Data Light	39
214	dbx translator	Desktop AI	180
119	R-Tree/C-Tree	FairCom	97
*	PC-Lint	Gimpel Software	124
113	Flowchart	HavenTree Software	20
148	C-Tree Query	Kurtzberg Computer System	175
160	Programming Tools	Lattice, Inc.	127
125	Epsilon	Lugaru Software	172
146	Help/Control	MDS, Inc.	163
259	386 Compiler	MetaWare	180
*	C Cross Compiler	Micro Tech Research	88
257	MACH 2	MicroHelp, Inc.	106
*	Windows Toolkit	Microsoft Corporation	74
191	Screenio	Norcom	164
222	Opt-Tech Sort	Opt-Tech Data Processing	4
101	Mace Utilities	Paul Mace Software	130
215	Periscope	Periscope Company	5
*	Mail Order	PC Brand	140-143
181	Instant-C	Rational Systems	109
201	BTrieve	Softcraft	2
142	Demo Program	Software Garden	93
168	Visible Computer	Software Masters	170
152	Turbo Professional	Sunnyhill Software	152
158	Taskview	Sunnyhill Software	152
115	Windows for C	Vermont Creative	17
	SOFTWARE UTILITIES		
254	Sidekick, Traveling Side Kick	Borland Int'l	Gatefold Cover & 1
254	SuperKey, Turbo Lightning	Borland Int'l.	Gatefold Cover & 1
254	Lightning Word Wizard	Borland Int'l.	Gatefold Cover & 1
114	Bookmark	Comcal	56
190	Command Plus	ESP Software	171
139	Easy Path	Isogon	112
191	Screenio	Norcom	168
101	Mace Utilities	Paul Mace	130
176	Speedstar	Storage Dimensions	164
	GRAPHIC SOFTWARE		
187	GRAPH C	Scientific Endeavors Corp.	125
	DATA BASE MANAGEMENT SOFTWARE		
153	Magic PC	Aker	158
254	Reflex, The Analyst	Borland Int'l.	Gatefold Cover & 1
254	Reflex Workshop	Borland Int'l.	Gatefold Cover & 1
211	Knowledgeman 2	Micro Data Base Systems	54
224	Clipper	Nantucket Corporation	139
166	Db Vista	Raima Corporation	11
144	DataEase	Software Solutions	100 & 101
156	Unify	Unify Corporation	133
	OPERATING SYSTEMS		
258	AMX Multitasking Exec.	KADAK Products Ltd.	180
174	MKS Tool Kit	Mortice Kern Systems, Inc.	161
	SECURITY DEVICES		
157	Software Sentinel	Rainbow Technologies, Inc.	180
	MAIL ORDER		
*	Mail Order	BC Associates	194
*	Mail Order	Business Engineering Syst.	192
112	Mail Order	Evsan Company	194
132	Mail Order	Hawaiian Village Computer	190
134	Mail Order	ITS	195
*	Mail Order	Microway	189
150	Mail Order	PC's Limited	12 & 13
175	Mail Order	Programmer's Connection	41-43
143	Util., Edit., Funct., Graphics	Programmer's Paradise	193
220	Mail Order	Programmer's Shop	46
162	Mail Order	Programmer's Shop	24 & 25
171	Mail Order	Ram Explosion	194
*	Mail Order	Scantel Systems Ltd.	194
184	Mail Order	Sunnytech, Inc.	190

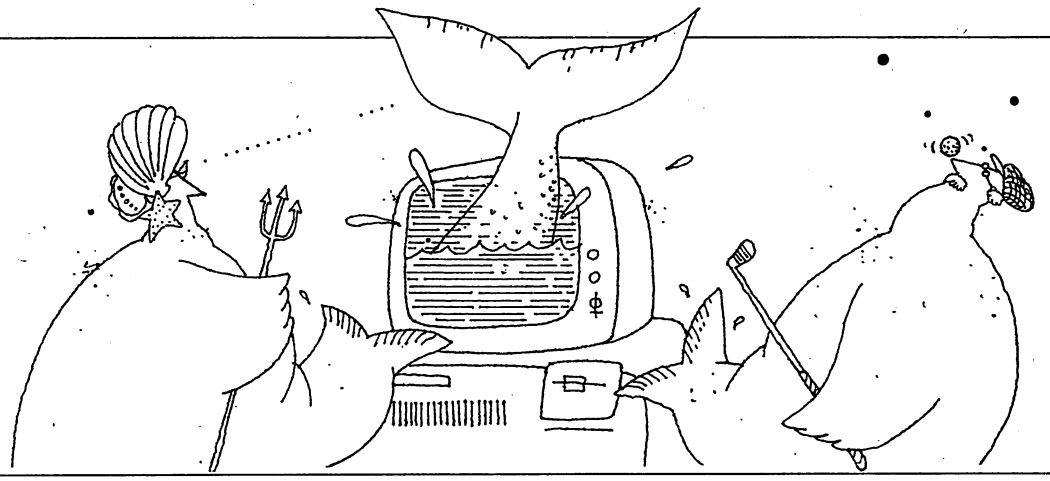


ILLUSTRATION • DAVID POVILAITIS

MARCH

March 15-20
GUIDE 67
Anaheim, CA
Sponsor: GUIDE International
Contact: Bill Reinberger, GUIDE International, 111 E. Wacker Drive, Chicago, IL 60601; 312/644-6610

March 17-19
Reliability in Distributed Software and Database Systems
Williamsburg, VA
Sponsor: IEEE-CS
Contact: Edwin C. Foudriat, NASA, Langley Research Center, Information Systems Division, MS 469, Hampton, VA 23665; 804/865-3535

March 17-19
Optical Storage of Documents and Images
Washington, DC
Sponsor: Rothchild
Contact: Rothchild Consultants, 256 Laguna Honda Blvd., San Francisco, CA 94116-1496; 415/681-3700

March 22-26
Computer Graphics '87
Philadelphia, PA
Sponsor: National Computer Graphics Association
Contact: NCGA, 2722 Merrilee Drive, Suite 200, Fairfax, VA 22031; 800/225-6422; in Virginia, 703/698-9600

March 23-27
Theory and Practice of Software Development (TAPSOFT '87)
Pisa, Italy
Sponsor: Università di Pisa
Contact: Pierpaolo Degano, Dipartimento di Informatica Università di Pisa, Corso Italia, 40 I-56100 Pisa, Italy

March 30-April 2
Ninth International Conference on Software Engineering
Monterey, CA
Sponsors: ACM SIGSOFT (Software Engineering) and IEEE-CS
Contact: William E. Riddle, Software Design and Analysis, 1760 Bear Mountain Drive, Boulder, CO 80303; 303/499-4782

March 31-April 3
ANSYS 1987 Conference and Exhibition
Newport Beach, CA
Sponsor: Swanson Analysis Systems, Inc.
Contact: Swanson Analysis Systems, Inc., Johnson Road, P.O. Box 65, Houston, PA 15342-0065; 412/746-3304

March 30-April 3
Robotics and Automation
Raleigh, NC
Sponsor: IEEE-CRA
Contact: Harry Hayman, 738 Whitaker Terrace, Silver Spring, MD 20901; 301/434-1990

APRIL

April 1-3
Database Systems for Office Automation, Engineering, and Scientific Applications
Darmstadt, West Germany
Sponsor: Gesellschaft für Informatik
Contact: H.-J. Schek, Technische Hochschule Darmstadt, Fachgebiet Datenverwaltungs-systeme I, Fachbereich Informatik, Alexanderstraβe 24, D-6100 Darmstadt, West Germany

April 5-9
CHI + GI '87
Toronto, Ontario, Canada
Sponsor: ACM
Contact: Wendy Walker, CHI + GI '87, Computer Systems Research Institute, University of Toronto, 2002-10 Kings College Road, Toronto, Ontario, Canada M5S 1A4; 416/978-5184

April 8-10
Mathematical Foundations of Programming Semantics
New Orleans, LA
Sponsor: ACM
Contact: ACM, 11 W. 42nd Street, New York, NY 10036; 212/869-7440

April 9-10
Advanced SPSS/PC+
Austin, TX
Sponsor: SPSS Inc.
Contact: SPSS Inc., Training Department, 444 N. Michigan Avenue, Chicago, IL 60611; 312/329-3557

April 9-10
A Manager's View of Expert Systems Building
Atlanta, GA
Sponsor: Georgia Institute of Technology
Contact: Deidre Mercer, Department of Continuing Education, Georgia Institute of Technology, Atlanta, GA 30332-0385; 404/894-2547

April 21-24
PERSCOMP '87
Sofia, Bulgaria
Sponsor: Bulgarian Academy of Sciences
Contact: Dr. Marcel Israel, ITKR/BAN, 113 SOFIA/BULGARIA, Acad. G.Bonchev Str., bl.2

April 22-24
AI '87
Long Beach, CA
Sponsor: Tower Conference Management
Contact: Jim Hay, Show Manager AI '87, TCM, 331 W. Wesley Street, Wheaton, IL 60187; 312/668-8100

April 27-29
Symposium on Security and Privacy
Oakland, CA
Sponsors: IEEE-CS and IACR
Contact: Virgil D. Gligor, Department of Electrical Engineering, University of Maryland, College Park, MD 20742; 301/454-8846

CALL FOR PAPERS

Deadline: March 30
International Conference on Information Systems
Pittsburgh, PA
(December 6-9, 1987)
Sponsors: Society for Information Management and the Institute for Management Sciences
Submit papers to: Charles H. Kriebel, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, PA 15213

FREE! FOR A LIMITED TIME ALR'S VIVID EGA WITH THE PURCHASE OF AN ALR DART OR DART Ø12 PC

Advanced Logic Research Gives You Everything a Mail Order House Can't. . .at the Same Price

See and Feel Your ALR Computer Before You Buy It.

If you've been thinking that the only way to get into a low-cost compatible PC is to make a blind purchase from a mail-order house, THINK AGAIN. Today's best value in Personal Computers comes from your local Advanced Logic Research Dealer! You can see and feel your ALR PC before you buy it. You can talk to an expert who can make qualified recommendations and take the time to custom-fit an ALR system to your particular needs.

Take a look at ALR's DART and DART Ø12 systems. DART offers the best 286 technology in a high performance, low-cost personal computer. The DART Ø12 blows the lid off of PC performance with true 12MHz speed and O wait state.

You Don't Need To Buy A Memory Card

The most intelligent PC designs to date, DART systems include LIM/EMS support (Lotus® /Intel® /Microsoft® Enhanced Memory Specification). That, plus having on board expandable memory of 2Mb (DART) and 3Mb (DART Ø12) means that you can break the 640K barrier without the added expense of a memory enhancement card.

Intelligence

Advanced Logic Research brings together the best features offered by anyone in the microcomputer industry. We provide a complete solution to the PC buyer's needs.

Intelligent design, quality conscious manufacturing and affordability along with the added value of local dealer support and guidance make the purchase of an ALR PC the best possible choice. Before you make a blind purchase from a mail-order computer warehouse see your local ALR dealer and take a long look at value.

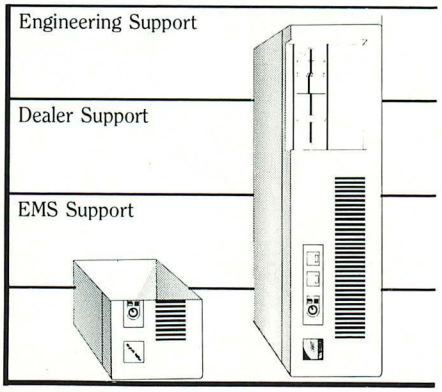
Performance

DART $2295.00

Phoenix BIOS
1.2Mb Floppy Drive
1Mb RAM Expandable to 2Mb
EMS Software
Hard/Floppy Disk Controller
2 Serial Ports/Parallel Port
Large RT Style Keyboard
Battery Backup Clock Calendar
System Set-up Configuration

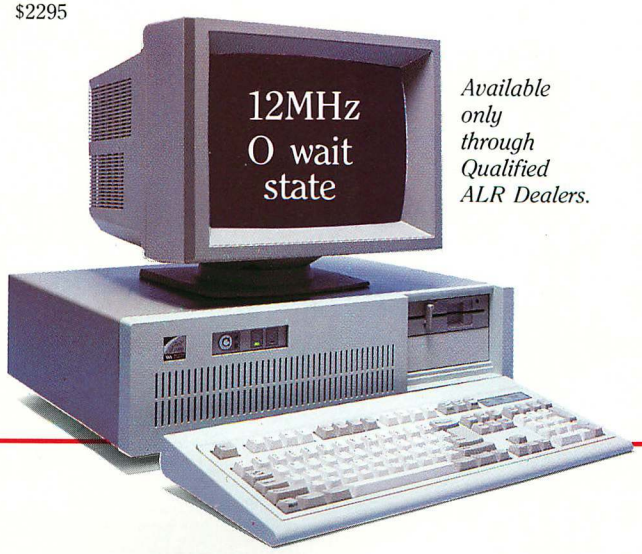
DART Ø12 $2695.00

Modular Bus Design
Phoenix Bios
80286-12 Microprocessor
80287 Socket on CPU Card
1.5Mb RAM Expandable to 3Mb
EMS Software
High Speed 80Ns 256K DRAM
1.2Mb Floppy Disk Drive
Combination Hard Disk/Floppy Controller
2 Serial Ports, 1 Parallel Port
Large RT Style Keyboard
Battery Backup Clock Calendar



Before you make a blind purchase, Think Again.





Available only through Qualified ALR Dealers.

Advanced Logic Research, Inc.
10 Chrysler, Irvine, California 92718 - (714) 581-6770
FAX: (714) 581-9240 - TELEX: 5106014525, Answer back Advanced Logic
Ask for ext. 12

CIRCLE NO. 116 ON READER SERVICE CARD

Lotus is a registered trademark of Lotus Development Corp.. Intel is a registered trademark of Intel Corp.. Microsoft is a registered trademark of Microsoft Corp.


No matter where you take CROSSTALK® Mk.4...
You won't encounter a PC communications program with as much versatility as CROSSTALK® Mk. 4. It has everything we could imagine you needing today. More protocols — X.PC, Xmodem, Kermit, and our own CROSSTALK. More terminal emulations, including complete IBM 3101, DEC VT-100, and TeleVideo 900 series. Concurrent communications capability — up to 15 sessions, each displayed in its own expandable window, or on separate "pages." Error checking at high speeds. Prepared script files to extract information from most popular information utilities. A powerful programming language to create customized scripts. Finally, we've built-in a bit of tomorrow. CROSSTALK Mk. 4 is based on a modular architecture that means we can add new capabilities by phone, as they come along. So you're getting more than today's standard in communications software. You're getting tomorrow's as well.
CROSSTALK® COMMUNICATIONS
Digital Communications Associates, Inc.
1000 Holcomb Woods Parkway
Roswell, Georgia 30076
CROSSTALK® Mk.4
CROSSTALK is a registered trademark of Digital Communications Associates, Inc. / DEC VT-100 is a registered trademark of Digital Equipment Corp
IBM is a registered trademark of International Business Machines Corp. / TeleVideo is a registered trademark of TeleVideo Systems, Inc. / X.PC is a trademark of Tymshare, Inc.
CIRCLE NO. 167 ON READER SERVICE CARD